



# INTERNATIONAL SCHOOL OF MANAGEMENT & TECHNOLOGY

## ASSIGNMENT COVER SHEET

### STUDENT DETAILS

<b>Student ID</b>		<b>Reg No.</b>	<b>M/615/1641</b>
<b>Family Name</b>		<b>Given Name</b>	<b>Saurav lamichhane</b>
<b>Enrolment Year</b>	<b>2018</b>	<b>Section</b>	<b>P5</b>
<b>Semester</b>	<b>3<sup>rd</sup></b>	<b>Email</b>	<b>saurav@ismt.edu.np</b>

### UNIT DETAILS

<b>Unit Title</b>	<b>Database management system</b>	<b>Unit Code</b>	<b>Y/615/1682</b>
<b>Assessor Name</b>	<b>Bikul Koirala</b>	<b>Issued Date</b>	<b>07 Sep. 2021 (re launch)</b>
<b>Assignment Title</b>	<b>Library Management System</b>		
<b>Assignment No</b>	<b>1/1</b>	<b>Submission Date</b>	<b>12/10/2021</b>
<b>Qualification</b>	<b>B.Tech HND</b>	<b>Campus</b>	<b>ISMT</b>

## **STUDENT ASSESSMENT SUBMISSION AND DECLARATION**

When submitting evidence for assessment, each student must sign a declaration confirming that the work is their own.

Student Name	Saurav Lamichhane	Assessor Name	Bikul Koirala
Issue Date	07 Sep. 2021 (re launch)	Submission Date	10 Dec 2021
Program	Database		
Unit Name	Database management system		
Assignment Title	Library Management System		

### **Plagiarism**

Plagiarism is a particular form of cheating. Plagiarism must be avoided at all costs and students who break the rules, however innocently, may be penalized. It is your responsibility to ensure that you understand correct referencing practices. As a university level student, you are expected to use appropriate references throughout and keep carefully detailed notes of all your sources of materials for material you have used in your work, including any material downloaded from the Internet. Please consult the relevant unit lecturer or your course tutor if you need any further advice.

### **Student Declaration**

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

Student signature: 

Date: 12/10/2021

# INTERNATIONAL SCHOOL OF MANAGEMENT AND TECHNOLOGY

KATHMANDU, NEPAL



Qualification	Unit Number & Title		
BTEC HND IN COMPUTING	Y/615/1682 – Unit 38: Database Management Systems		
Student Name	Assessor Name		
Saurav lamichhane	Bikul Koirala		
Assignment Launch Date	Due Date	Completion Date	Session/Year
07 September 2021(re-launch)	10 <sup>th</sup> Dec 2021		
Assignment Title	Assignment Number		
Library Management System	1/1		

## Assignment submission format

Each student has to submit their assignment as guided in the assignment brief. The students are guided what sort of information is to produce to meet the criteria targeted. You are required to make use of headings, paragraphs and subsections as appropriate, and all work must be supported with research and referenced using the APA referencing system.

## Learning outcomes covered

- LO1 Analyze different types of database management systems
- LO2 Design a database management system using a relational model to meet client requirements
- LO3 Develop a database management system using a suitable platform
- LO4 Demonstrate the system administration and management tools available on the chosen platform

## Scenario

“Oreland College” wants to reduce all the manual work. They want to convert all the registry things into digital format. As for now, they want to start with the Library section to reduce the crowd as well as for easy access.

They want the proficient and most easy system to manage all the operations of the Library department. As per their specifications, system analyst has collected following requirements:

- Any library member should be able to search books by their title, author, and subject category as well by the publication date.
- Each book will have a unique identification number and other details including a rack number which will help to physically locate the book.
- There could be more than one copy of a book, and library members should be able to check-out and reserve any copy. We will call each copy of a book, a book item.
- The system should be able to retrieve information like who took a particular book or what are the books checked-out by a specific library member.
- There should be a maximum limit (5) on how many books a member can check-out.
- There should be a maximum limit (10) on how many days a member can keep a book.
- The system should be able to collect fines for books returned after the due date.
- Members should be able to reserve books that are not currently available.
- Each book and member card will have a unique id. The system will be able to read book id from books and members' library cards.
- Email notification should be delivered on the available of the reserved books to those particular students.

Hence, the college requires a Library Management System that can perform all above tasks. Developer can have any Open source or vendor specific application as well as database.

#### **Keen features to be added:**

1. All the college members may not be “used to” the database directly. So, an application is required to operate. (web based or offline app)
2. Only few accesses should be created for Admin Section. (Admin user should have rights to create if required more)
3. Admin authorizations required to run all the operation. (Normal user can just view the database)
4. Backup system required at the backend.

#### **Assignment Task**

1. Write an article comparing and evaluating different database management systems and operating systems available (both open-source and vendor-specific), justifying the evaluation criteria. It should cover data models, modeling languages and specifically examine relational databases and their structures.
2. Design and develop a relational database management system and critically evaluate it against the client and system requirements.
  - The design will need to account for the key personnel involved in database management systems, their roles and responsibilities, and features of the systems that need to be managed, including data storage, backup, security, recovery and concurrency.
  - Give a clear indication of what the criteria will be for determining the development of any new database management system and show how the design will optimize system performance.
  - Using an open source language develop the fully functional database with User Interface and appropriate outputs. Populate the database with appropriate test data and fully test the system using appropriate tests, such as security, referential integrity, and functionality and use the testing outcomes to optimize it.
  - Write a report on your database, including design, development and testing, and critically evaluate the system against client and system requirements.
3. Demonstrate your database system to the Board. The IT Director is particularly interested in how to monitor and optimize system performance, as well as access audit logs, and the Compliance Director would like to see how security and authorizations are handled, especially in respect of GDPR. Assess the administration and maintenance tools used and identify areas for improvement and discuss potential further improvements to keep the system “future-proof”.

Pass	Merit	Distinction
<b>LO1 Analyze different types of database management systems</b>		
P1 Compare and contrast the different types of database models.	M1 Assess how relational database models and the process of normalization can provide reliable and efficient data structures.	D1 Critically evaluate different database management systems available in relation to open source and vendor specific platforms,

		justifying the criteria used in the evaluation.
<b>LO2 Design a database management system using a relational model to meet client requirements</b>	P2 Produce a design for a relational database management system to meet client requirements.	LO2 and LO3 D2 Critically evaluate the effectiveness of the system design and development against client and system requirements.
<b>LO3 Develop a database management system using a suitable platform</b>	M2 Analyze how the design will optimize system performance.  P3 Develop a fully functional system which meets client and system requirements, using an open source language (with an application software e.g. MySQL with front end Microsoft Access). P4 Test the system for functionality and performance.	
<b>LO4 Demonstrate the system administration and management tools available on the chosen platform</b>	M3 Implement effective features in the solution to handle concurrency, security, user authorizations and data recovery.	
P5 Demonstrate the tools available in the system to monitor and optimize system performance, and examine the audit logs. P6 Demonstrate the tools available in the system to manage security and authorizations.	M4 Assess the effectiveness of the system administration and management tools available on the platform identifying any shortcomings of the tools.	D3 Analyze any future improvements that may be required to ensure the continued effectiveness of the database system

## Grades Achieved

\_\_\_\_\_

**Note:** Refer the unit details provided in your handbook when responding all the tasks above. Make sure that you have understood and developed your response that matches the highlighted key words in each task.

### **Plagiarism Notice**

You are reminded that there exist **Academic Misconduct Policy and Regulation** concerning **Cheating and Plagiarism**.

#### **Extracts from the Policy:**

**Section 3.4.1:** Allowing others to do assignments / copying others assignment is an offence

**Section 3.4.2:** Plagiarism, using the views, opinion or insights / paraphrasing of another person's original phraseology without acknowledgement

### **Requirements**

- It should be the student's own work – **Plagiarism is unacceptable.**
- Clarity of expression and structure are important features.
- Your work should be submitted as a **well presented**, word-processed document with headers and footers, and headings and subheadings, **both in hard and soft copies**.
- You are expected to undertake research on this subject using books from the Library, and resources available on the Internet.
- Any sources of information should be **listed as references** at the end of your document and these sources should be referenced within the text of your document using **APA Referencing** style
- Your report should be illustrated with screen-prints, images, tables, charts and/or graphics.
- All assignments must be typed in **Times New Roman, font size 12, 1<sup>1/2</sup> spacing**.

**The center policy is that you must submit your work within due date to achieve “Merit” and “Distinction”. Late submission automatically eliminates your chance of achieving “Merit and Distinction”. Also, 80% attendance is required to validate this assignment.**

I declare that all the work submitted for this assignment is my own work and I understand that if any part of the work submitted for this assignment is found to be plagiarized, none of the work submitted will be allowed to count towards the assessment of the assignment.

<b>Assignment Prepared By</b> Bikul Koirala	<b>Signature</b>	<b>Date</b> March 10, 2021
<b>Brief Checked By</b> Dhruba Babu Joshi	<b>Signature</b> 	<b>Date</b> March 14, 2021

## Table of Contents

Part 1 .....	1
Introduction.....	2
Database .....	2
Database Management System: .....	3
Advantages of DBMS .....	3
Disadvantages of DBMS: .....	3
Flat File System .....	4
Database Model .....	5
Relational Database Model: .....	10
Properties of the relational database model .....	11
Structures of RDMS.....	11
Normalization: .....	14
Types of Normalization: .....	15
Justification .....	18
Comparison between DBMS and RDBMS .....	18
Advantages of Open Source Software: .....	20
Vendor Specific Software: .....	20
Evaluation of Open Source and Vendor Specific Software:.....	23
Conclusion .....	24
Part 2 .....	25
Introduction:.....	26
Entity Relationship Diagram: .....	26
Relationships between Entities: .....	27
ER Diagram Symbols: .....	27
ER Diagram for Library Management System .....	28
DFD (Data Flow Diagram): .....	28

Components of DFDs: .....	29
Database Schema .....	31
Class Diagram.....	32
Use Case Diagram.....	33
Design of Library Management System .....	36
Some important Contoller codes of LMS application .....	51
Database screenshots .....	55
Testing.....	57
User Interface testing (front-end).....	57
Database testing (back-end).....	57
Testing methodologies .....	57
Testing of system functionalities and performance .....	58
Overall system testing of Library management system:.....	58
Database Testing.....	66
Primary Key Constraint testing.....	66
Implementing effective features in LMS to handle concurrency, security, authorizations and data recovery.....	72
Database Maintenance .....	72
Database Security.....	76
Evaluation of effectiveness of designed system against client requirements .....	81
Benefits & Effectiveness of System.....	83
Conclusion .....	84
Introduction.....	86
System Monitoring and Optimizing Tools .....	86
Database Performance Monitoring Comparison:.....	87
Database Monitoring and optimization tool: .....	87
Monitoring and optimization in Library Management System.....	88

Extended Events.....	88
Error Log.....	94
SQL Server Profiler .....	96
Audit Logs .....	99
Activity Monitor .....	101
Indexes/Key: .....	103
Constraints: .....	104
Tools available in the system to manage security and authorizations .....	105
Database Backup:.....	106
Advantages of Database Backup: .....	108
Benefits of system authorization:.....	110
Restore Database:.....	110
System administration:.....	111
System management .....	112
Effectiveness of Database Management Tools .....	112
Advantages of database management tools .....	113
Stored Procedures .....	113
Database Backup & Recovery .....	114
Shortcomings of Database Tools .....	114
Further Enhancements on System.....	117
Conclusion .....	118
References.....	119

**Part 1.**

**Write an article comparing and evaluating different database management systems and operating systems available (both open-source and vendor-specific), justifying the evaluation criteria. It should cover data models, modeling languages and specifically examine relational databases and their structures.**

## DIFFERENT DATABASE MANAGEMENT SYSTEM & OPERATING SYSTEM AVAILABLE

The main purpose of this article is to compare the different types of database models. This article consist of relational database models and the process of normalization for reliable and efficient data structures. Finally, the article focuses on evaluation of different types of database management system available in relation to open source and vendor specification.

Saurav Lamichhane, ISMT College

### Introduction

The administration of “Oreland College” wants to reduce all the manual work. They want to convert all the registry things into digital format. As for now, they want to start with the Library section to reduce the crowd as well as for easy access. The college has been operating manually by keeping the record at register. However, with an increase in the number of students, the department is facing the problem to manage the flow of work. Thus, the college administration has decided to introduce the application in order to manage those records systematically and keep them safely for future use.

Database Management System is a software, which allows users to create, retrieve, update, delete and manage data in a database. Various DBMS software such as MySQL, Oracle, and SQL Server etc. are used for performing such operations on database.

In this work, I created a design for a relation database management system to satisfy the needs of the client. To provide an overview of the workflow of the intended system, diagrams such as the Entity Relationship (ER) diagram,

Data Flow Diagram (DFD), Class diagram, and Database Schema have been shown. A detailed assessment of how the design would improve system performance has been completed. A fully working system to suit the clients' and system requirements was constructed, and the system was evaluated for functionality and performance. In addition, I have built the solution's effective features for handling concurrency, security, user authorizations, and data recovery, as well as critically evaluating the system.

### Database

A database is a logically organized collection of structured data kept electronically in a computer system. A database management system is usually in charge of a database (DBMS). The data, the DBMS, and the applications that go with them are referred to as a database system, which is commonly abbreviated to just database. To make processing and data querying efficient, data in the most common types of databases in use today is often described in rows and columns in a sequence of tables. Data may then be accessed, managed, updated, regulated, and

organized with ease. For writing and querying data, most databases employ structured query language (SQL). To make processing and data querying efficient, data in the most common

### **Database Management System:**

A database-management system (DBMS) is a collection of linked data and a set of applications for accessing it. This is a database because it is a collection of related data with an inherent meaning. The database is a collection of data that provides information that is relevant to a business. The basic purpose of a database management system (DBMS) is to provide a convenient and efficient mechanism to store and retrieve database information. We use the term "data" to refer to known facts that can be recorded and have implicit meaning. Database systems are made to handle massive amounts of data. Data management entails both the creation of structures for storing data and the provision of methods for manipulating data. Furthermore, despite system crashes or efforts at illegal access, the database system must preserve the security of the information stored. If data is to be shared across multiple users, the system must avoid any unexpected outcomes.

### **Advantages of DBMS:**

Using a database management system for managing the data offers various kinds of advantages which are discussed below:

- As all the application program should be independent of the data

representation and storage, DBMS provides an abstract view of the data to insulate application code from such details.

- A DBMS utilizes a variety of sophisticated techniques to store and retrieve data efficiently. This feature is especially important if the data is stored on external storage devices.
- If data is always accessed through the DBMS, the DBMS can enforce integrity constraints on the data. Also, the DBMS can enforce access controls that govern what data is visible to different classes of users.
- A DBMS schedules concurrent accesses to the data in such a manner that users can think of the data as being accessed by only one user at a time. Further, the DBMS protects users from the effects of system failures.
- The use of Database management system largely reduces the time for the development of the application. The application also tends to be more robust.

### **Disadvantages of DBMS:**

There are some of the disadvantages as well when we work with the DBMS software which are listed below:

- For small and simple applications for single users a database system is often not advisable.
- A database system creates additional complexity and requirements. The supply and operation of a database management system with several users and databases is quite costly and demanding.
- The professional operation of a database system requires appropriately trained staff. Without a qualified database administrator nothing will work for long.

### **Flat File System**

A file system is a process of managing how and where data on a storage disk, which is also referred to as file management or FS. It is a logical disk component that compresses files separated into groups, which is known as directories. It is abstract to a human user and related to a computer; hence, it manages a disk's internal operations. Files and additional

directories can be in the directories. Although there are various file systems with Windows, NTFS is the most common in modern times. It would be impossible for a file with the same name to exist and also impossible to remove installed programs and recover specific files without file management, as well as files would have no organization without a file structure. The file system enables you to view a file in the current directory as files are often managed in a hierarchy.

A **flat file system** is a system of files in which every file in the system must have a different name. In Windows 95 and most other operating system today, files are managed in a hierarchical file system with a hierarchy of directories and subdirectories, each containing a number of files (or subdirectories). The operating system allows more than one file to have the same name as long as it is stored in a different directory. Early versions of the Macintosh and DOS operating systems used a flat file system.

## Difference between File system and DBMs

File System	DBMS
A file system is a software that manages and organizes the files in a storage medium. It controls how data is stored and retrieved.	DBMS or Database Management System is a software application. It is used for accessing, creating, and managing databases.
The file system provides the details of data representation and storage of data.	DBMS gives an abstract view of data that hides the details
Storing and retrieving of data can't be done efficiently in a file system.	DBMS is efficient to use as there are a wide variety of methods to store and retrieve data.
The file system doesn't have a crash recovery mechanism.	DBMS provides a crash recovery mechanism
Data inconsistency is higher in the file system.	Data inconsistency is low in a database management system
File System allows you to stores the data as isolated data files and entities.	Database Management System stores data as well as defined constraints and interrelation.
The centralization process is hard in File Management System	Centralization is easy to achieve in the DBMS system.
There is no efficient query processing in the file system.	You can easily query data in a database using the SQL language.

## Database Model

A database model is a type of data model that determines the logical structure of a database and fundamentally determines in which manner data can be stored, organized and manipulated.

**(SearchSQLServer 2019)** As the name implies, the data model focuses on what data is required and how it should be organized rather than what operations will be performed on the

data. The database model represents the logical structure of the database including the relationships and constraints. The database model are of various types and they are needed as per the information or the necessity. The data model, as the name suggests, concentrates on what data is needed and how it should be arranged rather than what operations will be performed on it. The database model depicts

the database's logical structure, including linkages and restrictions. Database models come in a variety of shapes and sizes, and they are required depending on the information or necessity.

There are various kinds of database models available as per the need of proper management and storage of data which are discussed below:

**1. Hierarchical Database Model:** The database model which organizes the data into a tree like structure is known as the hierarchical database model. It is the model where all the data are linked and organized into a single root. The hierarchy starts from the root data and expands like a tree, adding child nodes to the parent nodes. There is the presence of one-to- many relationship between two different types of data. There is a child node which will only have a single parent node. For instance: windows registry used in windows XP.

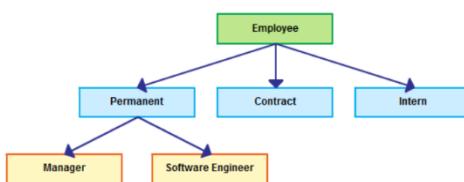


Figure: Hierarchical model

The advantages of Hierarchical database model are listed below:

- In hierarchical model, the data at the top can be accessed easily and quickly.

- The model allows easy addition and deletion of new information.

**2. Network Database Model:** “Network model has the entities which are organized in a graphical representation and some entities in the graph can be accessed through several paths” (Thakur, 2016). It is an extension of hierarchical database model where the data are organized more like a graph and are allowed to have more than one parent node. Data is more related as more relationships are established in this database model. This model is used for mapping many to many data relationships.

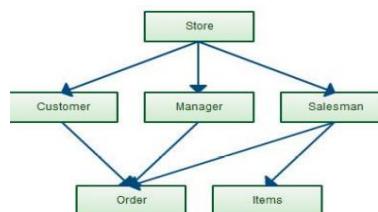


Figure: Network Database Model

The advantages of network database model are listed below:

- The network model can represent redundancy in data more effectively than in the hierarchical database model
- The model is simple and easy to design.
- In network database model, accessing data is easier and

flexible than the hierarchical model.

- Isolation of program from the physical storage details is easier and better in network database model.
- The network model can handle the one to many and many to many relationships which is real help in modelling the real-life situation.

The disadvantages of network database model are listed below:

- In network database model, the structural changes to the database is very difficult.
- The whole database structure becomes very complex as all the records are maintained using pointers.
- The large number of pointers adjustments are necessary during the insertion, deletion and updating operations of any record.

### 3. Relational Database Model:

Relational model is the most popular model and the most extensively used model. In this model the data can be stored in the tables and this storing is called as relation, the relations can be normalized and the normalized relation

values are called atomic values. Each row in a relation contains unique value and it is called as tuple, each column contains value from same domain and it is called as attribute. In relational model, data is organized in rows and column structure which is two-dimensional tables and the relationship is maintained by storing a common

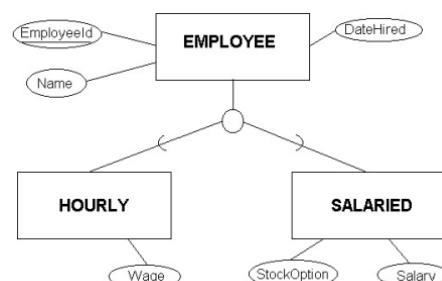


Figure: Relational Database Model

The advantages of relational database model are listed below:

- It is easier to use and maintenance of security is easier compared to other database models.
- Changes in the database structure do not affect the data access in this model.
- The database design, maintenance, administration and usage are much easier than in this model than other models as the model supports both data independence and structure independence concept.
- Using of complex query to access or modify the data from database is possible in this model.

- The information can be revised easily tables consisting of rows and columns is much easier to understand.

The disadvantages of relational database model are listed below:

- Relational model is not suitable for huge database but suitable for small database.
- Mapping of objects is difficult in relational database as object oriented paradigm is missing.
- Relational database system hides the implementation complexities and the physical data storage details from the users.
- Ease of design can lead to bad design and hardware overheads are incurred which make it costly.

#### 4. Object Oriented Database Model:

Object oriented data model is one of the developed data model and this can hold the audio, video and graphic files. These consist of data piece and the methods which are the DBMS instructions. In object oriented model,

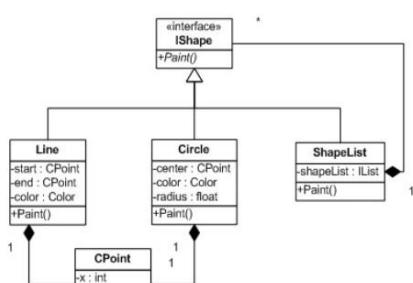


Figure: Object Oriented Database Model

data is represented in the form of objects, as used in OOP. The key features of object oriented paradigm such as encapsulation, inheritance abstraction, polymorphism etc. are used in this model. An object-oriented database stores complex data and relationships between data directly, without mapping to relational rows and columns.

The advantages of object oriented database model are:

- Object oriented databases provide code reusability, real world modelling and improved reliability and flexibility.
- The object-oriented database has low maintenance costs as compared to other model due to encapsulation and reusability.
- Unlike other database models, object oriented model can handle the different types of data, for example, pictures, voice video, including text, numbers and so on.

The disadvantages of object oriented database model are:

- The use of object oriented model is relatively limited compared to relational database model.

- There is no adequate security mechanisms in object oriented model.
- The object oriented model is complex than other database model.
- Object oriented model lacks a universally defined data model.

**The differences between hierarchical, relational and object oriented database model** on various basis are explained below:

- **Definition:** A hierarchical model is a data structure that uses parent-child relationships to arrange data in a tree-like structure, whereas a network model is a database model that allows several records to be linked to the same owner file. A relational model, on the other hand, is a database model for managing data in the form of tuples that are organized into relationships. Data is arranged in an object-oriented paradigm as objects, which are instances of classes.
- **Basis:** Hierarchical model arranges data in a tree similar structure while network model organizes data in a graph structure. In contrast, relational model arranges data in tables. The object oriented data in objects.
- **Relationship:** Furthermore, a key distinction between a hierarchical

network and a relational database model is that a network model depicts many to many relationships, whereas a hierarchical model represents one to many relationships. A relational model can also depict one-to-many and many-to-many relationships. The data has a direct relationship with the object-oriented model.

- **Accessing data:** Although it is difficult to access data in the hierarchical model, it is easier to access data in the network model, relational model and object oriented model.
- **Flexibility:** Another difference in database models is their flexibility. The hierarchical model is less flexible, but the network model and relational model are flexible. Object oriented model is very flexible than other data models and also has the feature of reusability.
- **Usability:** The hierarchical model is a classic model that is rarely utilized nowadays. More people use the network model than the hierarchical model. Due to its simplicity, the relational database model is the most extensively used. Despite the fact that the object-oriented database model offers many features, it is only employed infrequently. lesser than relational model due to its complexity and security.

Hierarchical Model	Network Model	Relational Model	object oriented model
A structure of data organized in a tree like model using parent, child relationships.	A database model that allows multiple records to be linked to the same owner file.	A database model to manage data as tuples grouped into relations (tables).	A database model to manage data as objects which is instance of class
Arrange data in a tree similar structure	Organizes data in a graph structure	Arrange data in tables	Arrange data in objects
Represents "one to many" relationship.	Represents "many to many" relationship.	Represents both "one to many" and "many to many" relationship.	Represents "many to many" relationship.
Difficult to access data	Easier to access data	Easier to access data	Easier to access data
Less flexible	flexible	flexible	flexible

### Relational Database Model:

The relational model is the most widely used and most often utilized model. The data in this model can be kept in tables, which is referred to as a relation. The relations can be normalized, and the normalized relation values are referred to as atomic values. Each row in a relation is referred to as a tuple, and each column is referred to as an attribute, as each column contains values from the same domain. The data in a relational model is structured in two-dimensional tables with rows and columns, and the relationship is maintained by storing a shared field. The advantages of relational database model are listed below:

- It is easier to use and maintenance of security is easier compared to other database models.
- Changes in the database structure do not affect the data access in this model.
- The database design, maintenance, administration and usage are much easier than in this model than other models as the model supports both data independence and structure independence concept.
- Using of complex query to access or modify the data from database is possible in this model.

- The information can be revised easily tables consisting of rows and columns is much easier to understand.

### Properties of the relational database model

#### Properties of Relational Tables:

- Data is presented as a collection of relations.
- Each relation is depicted as a table.
- Columns are attributes that belong to the entity modeled by the table (ex. In a student table, you could have name, address, student ID, major, etc.).
- Each row ("tuple") represents a single entity (ex. In a student table, John Smith, 14 Oak St, 9002342, Accounting, would represent one student entity).
- Every table has a set of attributes that taken together as a "key" (technically, a "super key") uniquely identifies each entity (Ex. In the student table, "student ID" would uniquely identify each student – no two students would have the same student ID)

### Structures of RDMS

- Tables:** In Relational database model, a table is a collection of data elements organized in terms of rows and columns. A table is also considered as a convenient representation of relations. But a table can have duplicate row of data while a true relation cannot have

duplicate data. Table is the most simplest form of data storage. Below is an example of an Employee

ID	Name	Age	Salary
1	Hari	23	15,000
2	Shyam	27	20,000
3	Man bhadur	28	18,000

- Tuple:** A single entry in a table is called a Tuple or Record or Row. A tuple in a table represents a set of related data. For example, the above Employee table has four tuples/records/rows.

1	Hari	23	15,000
---	------	----	--------

- Attribute:** A table consists of several records (row), each record can be broken down into several smaller parts of data known as Attributes. The above Employee table consist of four attributes, ID, Name, Age and Salary.



- Relation Schema:** A relation schema describes the structure of the relation, with the name of the relation (name of table), its attributes and their names and type.

#### 4. Relational Integrity Constraints:

Every relation in a relational database model should abide by or follow a few constraints to be a valid relation, these constraints are called as Relational Integrity Constraints. The Four main Integrity Constraints are:

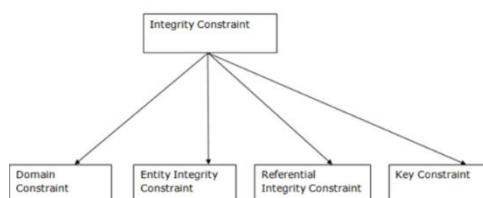


Figure: Relational Integrity Constraint

i) **Key Constraints:** We store data in tables, to later access it whenever required. In every table one or more than one attributes together are used to fetch data from tables. The Key Constraint specifies that there should be such an attribute(column) in a relation(table), which can be used to fetch data for any tuple(row). The Key attribute should never be NULL or same for two different row of data. For example, in the Employee table we can use the attribute ID to fetch data for each of the employee. No value of ID is null and it is unique for every row, hence it can be our Key.

#### Example

ID	NAME	SEMESTER	AGE
1000	Tom	1 <sup>st</sup>	17
1001	Johnson	2 <sup>nd</sup>	24
1002	Leonardo	5 <sup>th</sup>	21
1003	Kate	3 <sup>rd</sup>	19
1004	Morgan	8 <sup>th</sup>	A

The **Highlighted ID** Shouldn't be allowed because all row must be unique.

ii) **Domain Constraints:** Domain constraints refers to the rules defined for the values that can be stored for a certain attribute. Like we explained above, we cannot store Address of employee in the column for Name. Similarly, a mobile number cannot exceed 10 digits.

#### Example

ID	NAME	SEMESTER	AGE
1000	Tom	1 <sup>st</sup>	17
1001	Johnson	2 <sup>nd</sup>	24
1002	Leonardo	5 <sup>th</sup>	21
1003	Kate	3 <sup>rd</sup>	19
1004	Morgan	8 <sup>th</sup>	A

The **Highlighted part** isn't allowed because age is integer attribute

iii) **Entity integrity constraints:** The entity integrity constraint states that primary key value can't be null. This is because the primary key value is used to identify individual rows in relation and if the primary key has a null value, then we can't identify those rows. A table can contain a null value other than the primary key field.

#### Example:

EMP_ID	EMP_NAME	SALARY
123	Jack	30000
142	Harry	60000
164	John	20000
	Jackson	27000

The **highlighted part** doesn't contain the null value as it contains primary key.

**Relationships in DBMS:** Any association between two entity types is called a relationship. Entities take part in the relationship. It is represented by a diamond shape. **For example,** A teacher teaches students. Here, "teaches" is a relationship and this is the relationship between a Teacher entity and a Student entity. We have two entity types of 'Customer'(Customer\_id, Name, City, Phone) and 'Account'(Account\_no, Type, Balance). We store the data of 'Customer' in one table and his accounts details in the 'Account' table. Now, to link these two tables we need to insert the primary key 'Customer\_id' of the 'Customer' table in the 'Account' table. This key acts as a foreign key for the 'Account' table and refers to a column with the same name in the 'Customer' table. This is how a relationship between two tables is established. There are three types of relationships that can exist between two entities.

- One-to-One Relationship
- One-to-Many or Many-to-One Relationship
- Many-to-Many Relationship

**One-to-One Relationship:** Such a relationship exists when each record of one table is related to only one record of the other table. For example, If there are two entities 'Person' (Id,

Name, Age, Address) and 'Passport'(Passport\_id, Passport\_no). Therefore, each person can have only one passport and each passport belongs to only one person.



Figure:One-to-One Relationship

**One-to-Many or Many-to-One Relationship:** Such a relationship exists when each record of one table can be related to one or more than one record of the other table. This relationship is the most common relationship found. A one-to-many relationship can also be said as a many-to-one relationship depending upon the way we view it. For example, If there are two entity type 'Customer' and 'Account' then each 'Customer' can have more than one 'Account' but each 'Account' is held by only one 'Customer'. In this example, we can say that each Customer is associated with many Account. So, it is a one-to-many relationship. But, if we see it the other way i.e many Account is associated with one Customer then we can say that it is a many-to-one relationship.



Figure:One-to-Many Relationship

**Many-to-Many Relationship:** Such a relationship exists when each record of the first table can be related to one or more than one record of the second table and a single record of the second table can be related to one or more than one record of the first table. A many-

to-many relationship can be seen as a two one-to-many relationship which is linked by a 'linking table' or 'associate table'. The linking table links two tables by having fields which are the



Figure: Many-to-Many Relationship

### Normalization:

“Database Normalization is a technique of organizing data in a systematic way of decomposing tables to eliminate data redundancy and undesirable characteristics such as Insertion, Update and Deletion anomalies” (**Studytonight, 2016**). Database normalization usually means to eradicate all the anomalies or flaws and make the database consistent. By normalizing database, we can arrange data into tables and columns ensuring each table contains only related data. If data is

Employee ID	Employee Name	City	Department
1	Saurav Lamichhane	Kathmandu	IT
2	Abishek Jha	Maheendranagar	HR
3	Anish rawal	Dhangadi	Finance

not directly related, we have to create a new

table for the data. For example, if there is a customer table we would normally create product table for the number of product they have order another table is created to store each order made by customer which is linked by their primary key which helps to easily update, delete and search relevant data in database across table. It works to remove following anomalies:

Let's take a real life example of data (Employee Table) which is not normalized:

**Update Anomaly:** If we wish to change the city of an employee who appears twice or more in the table, we must change the city of all the employees. As a result, data will become ambiguous. For example, Employee named ‘Saurav’ is associated with two departments, both of which have non-Automic values.

**Insertion Anomaly:** If a new employee hasn't been assigned to a department yet. As a result, we must insert a null value, causing an insertion anomaly.

**Deletion Anomaly:** If Employee id 1 has only one department and that temporarily is destroyed, the whole student record is also deleted.

To get around these problems, we'll need to employ database normalized forms. When attempting to normalize a database, keep the following four things into consideration:

- Data is organized into logical groups.
- Reduce the amount of data that is duplicated.

- Organize the data in such a manner that when changes are needed, they only need to be made in one place.
- Data can be accessed and manipulated quickly and effectively by the user

### **Types of Normalization:**

There are various kinds of normalization to remove the flaws and organize the database in a systematic way. Let us consider the club wants to keep record of the teams, players. The database was designed as shown below:

As we can observe that there are various flaws in this schema such as multiple values in single cell, data redundancy and duplication. The different forms of normalization can be used to avoid these flaws which are explained below:

- **First Normalized Form(1NF):** 1 NF states that, “all the attributes in a relation must have atomic domains. The values in an atomic domain are indivisible units.” It means that the attribute cannot have multiple values and single values must be kept for each cell of the table. In 1 NF following activities are done:
  - i. There should be single values for each cell or column.
  - ii. Repeating groups are eliminated in individual tables.
  - iii. For each set of related data, a separate table is created.

iv. Each set of related data is identified with a primary key.

The table above depicts that the players ‘Cristiano Ronaldo and Sergio Ramos’ are kept in same cell and ‘Danilo and Vincent’ are also kept in the same cell which is against the First Normalization Form. The table depicted can be normalized in 1 NF as follows:

SN	Team Name	Player Name	Competition Name
1	Real Madrid	Christiano Ronaldo	Laliga
2	Real Madrid	Sergio Ramos	Laliga
3	Barcelona	Messi	Laliga
4	Manchester City	Danilo	EPL
5	Manchester City	Vincent	EPL

Hence, the above table drawn fulfils the condition of 1 NF and it is made sure that atomic values are available in single cell.

- **Second Normalized Form (2 NF):** 2 NF states that, “it should meet all the rules for 1NF and all non-key attributes must fully depend on the primary key.” The above table is not in 2 NF as there are many data redundancy. The team name ‘Real Madrid’ and ‘Atletico Madrid’ are repeated twice. These redundancy creates delete and update anomalies. For instance: if we need to change the competition name of ‘Real Madrid’ then we need to update it in two rows. In the above table, SN, Team and Players are “Prime Attribute” whereas Competition name is a non-

prime attribute. According to 2 NF, competition name must completely depend on those prime attributes but it does not depend on the prime attributes wholly. The table depicted can be normalized into 2 NF by:

Team Name	Competition Name
Real Madrid	Laliga
Real Madrid	Laliga
Barcelona	Laliga
Manchester City	EPL
Manchester City	EPL

- **Third Normalized Form(3NF):** The statement of Third normalization form is “it should meet all the rules of 2 NF and there should not be transitive functional dependency.” To clearly illustrate the statement, competition id is added to the table which is as shown below:

Team Name	Competition Name	Competition id
Real Madrid	Laliga	1
Barcelona	Laliga	1
Manchester City	EPL	2

If we examine the table, it is in 2 NF as ‘Competition name’ and ‘Competition id’ both are nonprime attributes and depend on the prime attribute ‘Teams.’ But it is not yet in 3 NF as non-prime attribute ‘Competition name’ does not directly depend on the prime attribute

‘Teams’. Rather if Competition id is known, competition name can be found easily. So, the attribute, Competition name is actually dependent on the prime attribute through the Competition name(transitive dependency). The 3 NF data is shown below:

Competition Name	Competition id
Laliga	1
Laliga	1
EPL	2

SN	Team Name	Player Name
1	Real Madrid	Christiano Ronaldo
2	Real Madrid	Sergio Ramos
3	Barcelona	Messi
4	Manchester City	Danilo
5	Manchester City	Vincent

Hence, by using the three normalized form, the data are arranged in systematic way eliminating the data redundancy, anomalies and dependencies and makes easier understanding of the data.

- **BCNF:** Boyce and Codd Normal form are a higher form of third normal form. This form deals with the certain types of anomalies which is not handled by 3NF. table which does not have multiple overlapping candidate keys is said to be in BCNF. For a table to be in BCNF, following conditions must be satisfied:

- i. R must be in third normal form
  - ii. And there should be a super key for each functional dependency ( $X \rightarrow Y$ )
- **4NF:** We can call a database table to be a fourth normal form when it follows the following criteria:  
It is in the Boyce-Codd Normal Form.  
And, it doesn't have Multi-Valued Dependency.

Database is the organized collection of data that are stored and access electronically from a computer system. For the storing of data from the generation different types of database like network, hierarchical, relational database are used on the basis of requirement of the project.

**(Tutorialspoint.com 2019)** Despite the fact that there are various types of database management, I favor relational database administration for my project. As the name implies, a relational database stores and allows access to data points that are connected to one another. The logical data structures, such as data tables, views, and indexes, are separated from the physical storage structures in the relational paradigm. The security of RDMS is one of the reasons for its selection. Each database has its own set of advantages in its own field. Data is kept in numerous tables in RDMS, making it easier for users to access data. The base of forming relational database table is normalization so that it avoids data redundancy and data duplication in database.

- **Data Model:** Data model is the logical structure of a database, defining how data is connected to each other and how they are processed and stored inside the system. For that type of modeling RDMS provide clear view of data flow in the system.
- **Data consistency and data security:** The most valuable thing for any project is providing data security. In the term of RDMS data are encrypted and higher security like data encapsulation, data protection, data hiding is used for protection of data.
- **Manageability:** For starters, an RDB is easy to manipulate. Each table of data can be updated without disrupting the others. We can also share certain sets of data with one group, but limit their access to others such as confidential information about employees.
- **Flexibility:** If you need to update your data, you only have to do it once – so no more having to change multiple files one at a time. The primary key and the foreign key are used to link the tables together. And it's pretty simple to extend your database. If your records are growing, a relational database is easily scalable to grow with your data.
- **Avoid Errors:** There's no room for mistakes in a relational database because it's easy to check for mistakes

against the data in other parts of the records. And since each piece of information is stored at a single point, you don't have the problem of old versions of data clouding the picture.

- **Implementation of service cost:** As RDMS is used for higher level project and middle level project which budget is higher for the completion of the project. So, the implementation of RDMS cost is higher.

### **Justification**

Relational database management is the better and right database system for any kind of project because the data are stored in the form of structured data in rows and columns. If we use RDMS for the data storage we can have more security of data. In the RDMS the data that need to be stored are stored in the table. By the use of RDMS we can create multiple users. The main fact of using RDMS for storing of data is due to ACID (Atomicity, Consistency, Isolation and Durability). ([Tutorialspoint.com](http://Tutorialspoint.com) 2019)

### **Comparison between DBMS and RDBMS**

DBMS	RDBMS
DBMS applications store data as file.	RDBMS applications store data in a tabular form.
Data is generally stored in either a hierarchical form or a navigational form.	The tables have an identifier called primary key and the data values are stored in the form of tables.
Normalization is not present.	Normalization is present.
Does not support distributed database.	Supports distributed database.

**Open source software:** Open source software is software in which the source code is freely provided with the ability to modify it, on the condition that redistribution is not restricted, and is available for no more than the cost of production. The source code for open source software can be obtained on the internet for free. Other organizations or users can copy, modify, or delete the codes. The following is a list of some of the most popular open source software:

- **Maria DB:** Maria DB Server is one of the most widely used database servers. It was created by the original MySQL developers and is promised to remain open source. In a wide range of applications, from banking to internet, Maria DB converts data into structured information. It's a better-than-MYSQL drop-in alternative. Maria DB is utilized because it is fast, scalable, and reliable, and because it has a large ecosystem of storage engines, plugins, and other tools that make it very versatile for a wide range of use cases. Maria DB is a relational database that uses a SQL interface to access data. It is developed as open source software. GIS and JSON features have been added to Maria DB in recent editions. MariaDB's reputation as a database system is bolstered by a number of features. One of its most notable characteristics is its speed.

MariaDB is incredibly scalable, with the ability to manage millions of tables and billions of rows of data. It can also swiftly and smoothly manage tiny volumes of data, making it ideal for small businesses or personal projects. Another aspect that distinguishes it from its predecessors is its emphasis on security. The storage engine, which manages queries and interfaces between a user's SQL statements and the database's back-end storage, is the most important piece of software in any database management system. MariaDB provides a number of storage engines, each with its own set of benefits.



- **MySQL:** MySQL is a popular free and open-source database management system. Oracle Corporation created it, and it was written in C and C++. It, like all other database systems, is a relational database system. It supports SQL, or Structured Query Language, which is a standard language for querying and updating data, as well as managing and administering it. MySQL runs on a wide range of operating systems. Apple Mac OS X, Microsoft

Windows, Linux, and a variety of UNIX variations are just a few examples. It is a fast database management tool that uses triggers and views to enforce business rules and only displays data that the user need. MySQL has a client-server architecture. MySQL's fundamental component is the MySQL server, which manages all database commands (or commands). The MySQL server is available as a standalone program for usage in a client-server networked environment, as well as a library that may be integrated (or linked) into other programs. MySQL was created with the goal of quickly processing big databases. Despite the fact that MySQL is normally installed on a single system, it can transfer the database to several places because users can access it through various MySQL client interfaces. These interfaces communicate with the server using SQL commands and then show the results.



### **Advantages of Open Source Software:**

The advantages of open source software are discussed below:

- Source code availability makes open source software possible to perform a thorough inspection and verify the correctness of the algorithm and the implementation scheme used.
- Open source software usually does not need as many service packs, updates and reduce the maintenance cost.
- No per-copy fees can be asked for modified versions, and anyone can use the current code base to start new projects.
- The availability of the source code and the tight to modify it enables the unlimited tuning and improvement of a software product.
- There is no one with the power to restrict in a unilateral way how the open source software is used, even in a retroactive way.

### **Disadvantages of Open source software:**

The disadvantages of open source software are discussed below:

- There is no guarantee that open source software development will happen as a project will ever reach a usable stage is not known, and even if it reaches it, it may die later if there is not enough interest.

- There may be a significant problems connected to intellectual property.
- Documentation and user manuals can be hard to follow for non-techies.
- Customization costs development time and money as there are limited financial incentives for improvements and innovations.
- Most business operate on vendor specific software, so sharing information or documents might be difficult.

### **Vendor Specific Software:**

Vendor specific software is software that has been designed and coded by or for a specific person, organization, or collection of organizations who own or control the software's intellectual property rights. This type of software is typically designed to a firm's specifications, used solely by that company, and not for sale to third parties or the general public. The software's user has complete control over the code and how it is used. Its use is almost always subject to severe limitations, and its source code is almost always kept hidden. Below is a list of some of the most popular vendor-specific applications.

- **Oracle:** Oracle which is also known as oracle DB or Oracle database is a relational database management system. It is produced and marketed by Oracle Corporation. “Oracle database is

the first database designed for enterprise grid computing. The enterprise grid computing provides the most flexible and cost effective way to manage information and applications" (javaTpoint, 2011). Oracle is well-known in the business world because of its numerous features. Oracle databases are updated regularly, with new and better features aimed at enterprises and organizations. Oracle has risen to the top of the software market as a result of their unwavering commitment to providing the best databases. As a result, there are various advantages to adopting Oracle databases. Oracle databases have the advantage of Oracle's commitment to client satisfaction. All Oracle databases, for example, are backward compatible. This enables companies to improve their systems without having to revamp their database system entirely. This provides efficient and low-cost updates. Another important advantage offered by Oracle databases are their reliability. Oracle is a database that delivers excellent performance when challenged with demanding tasks. The ACID test, which is an important tool used to ensure the integrity of data stored, was easily passed by Oracle databases. This test is important since reliable data

storage is the main purpose of a database. Oracle databases incorporate Flashback technology, which is a significant advantage. In the event of an application outage, due to any number of reasons, it is important not to lose data stored on a database system. Oracle's Flashback technology allows for efficient recovery of data incorrectly deleted or lost. Oracle databases have also successfully featured the four properties that all database systems must have. These four properties are atomicity, consistency, isolation and durability. All four of the properties are well maintained by Oracle databases, thus providing a reliable and competent database system.



- **Microsoft SQL Server:** Microsoft SQL Server is a relational database management system that Microsoft has created. The SQL Server's primary job as a database server is to store and retrieve data for other applications. "Microsoft SQL Server, like other RDBMS software, is based on SQL, a standardized programming language used by database administrators (DBAs) and other IT professionals to

administer databases and query the data they contain. SQL Server is bound to Microsoft's Transact-SQL (T-SQL), a SQL implementation that adds a set of proprietary programming extensions to the standard language" (Rouse, 2019). MS SQL installation has been simplified. It is installed via a setup wizard, and the needed updates are automatically identified and downloaded by the installer. Because updates are installed automatically, the complexity of installing the software is greatly reduced. After that, further components like analytical and database services can be added individually. Automatic updating also saves a lot of money on maintenance. MS SQL Server features a transparent data compression function as well as encryption built-in. To encrypt data, users do not need to make any changes to their programs. Access control and efficient permission management technologies are available on the MS SQL server. Furthermore, when it comes to data collection, it performs better. Along with disk partitioning, SQL Server features powerful data management and data mining tools. Following appropriate data management procedures helps assure the server's optimal maintenance. These

procedures also assist us in ensuring data availability and recoverability. SQL server's security features are also excellent. SQL Server 2008, for example, includes improved password management capabilities to enforce stronger passwords and frequent password changes, as well as strong authentication and access security. Policy-Based Management in SQL Server 2008 detects non-compliance security policies, allowing only authorized workers access to the database. Security audits and events can be written to log files automatically.

#### **Advantages of Vendor Specific Software:**

The vendor specific software offers following advantages which are discussed below:

- The vendor specific software model provides the vendor a guaranteed income which can be used to better service their customers.
- A vendor specific company usually needs to listen to the needs of their customers, and respond and develop accordingly.
- It guarantees structured innovation, which is innovation that is planned within a single responsible organization.
- Support and speed of response to problems is available as the company

can usually fix it more quickly with the help of consultant than waiting for developer's community.

- Vendor specific software licenses provides protection for intellectual property.

#### **Disadvantages of Vendor Specific Software:**

There are some of the disadvantages as well which are discussed below:

- It is expensive to license and maintain these software.
- It has little or no local support and if the software manufacturer decides to discontinue development of the product, no one has the right to take the program and continue development on it.
- The quality of the software depends entirely on the owner organization that can cause problem if the perspective and resources of owner and using organizations are different.
- The high cost of the vendor specific software leads to illegal copying of the software.

#### **Evaluation of Open Source and Vendor Specific Software:**

The five basic aspects of evaluation of open source and vendor specific software are discussed below:

- 1) **Cost:** Open source software is frequently referred to as free software.

Extras such as assistance, additional services, or more functionality may, however, come at a cost. Vendor-specific software is almost always a premium product. The price will vary depending on the software's complexity. While the price may be more, we get a better product, complete support, functionality, and innovation in exchange. Most organizations, on the other hand, provide free trials to persuade customers that their product is the correct fit.

2) **Service:** In terms of service quality, vendor-specific software reigns supreme. Its fees include the ability to contact assistance and, in most situations, receive a response within one business day. The response is well-documented and structured. Such a choice is not available for open source software. Forums, useful publications, and a professional expert are the sole sources of assistance. However, it is unsurprising that we will not receive a high degree of response if we use such a service.

3) **Source code availability:** Open source software allows users to modify the source code without restriction. Individual users can create what they desire while also benefiting from the ingenuity of others in the user

community. Because the source code is readily available, software engineers can easily improve existing programs. Because the source code of vendor-specific software cannot be altered or read, it is more restricted than open source software. Such limitations, on the other hand, may add to the security and reliability of vendor-specific software.

- 4) Usability:** User guidelines for open source software are intended for developers rather than non-techies. Even the standards and structure of those documents are not followed. Usability is one of the benefits of closed source software. Typically, documentation is well-written and includes explicit instructions.
- 5) Security:** The community can access, distribute, and modify the code of open source software, which means that anybody can patch, upgrade, and test the broken code. After each release, the defects are rapidly resolved and the code is extensively reviewed. The source code, on the other hand, is free

for hackers to practice on due to its accessibility. Vendor-specific software, on the other hand, can only be fixed by a vendor. If something goes wrong with the software, we send a request to the support team and wait for a response. When compared to open source software, solving the problem can take a lengthy time.

### Conclusion

The different database models were thoroughly examined and compared. The comparison of the different database models provided the result that relational database model is widely in practice due to its simplicity nature. The object oriented database model has not fostered yet though it has wide range of features. The selection of database models entirely depends upon the needs and requirements. The normalization techniques and its types were explained that helped to eliminate various redundancies and anomalies present in the database. Moreover, various open source and vendor specific database managements system were presented that provided detail explanation of those systems and platforms.

**Part 2**

- **Design and develop a relational database management system. and analyze how the design will optimize system performance.**
- **The design will need to account for the key personnel involved in database management systems, their roles and responsibilities, and features of the systems that need to be managed, including data storage, backup, security, recovery and concurrency.**
- **Give a clear indication of what the criteria will be for determining the development of any new database management system and show how the design will optimize system performance.**
- **Using an open-source language develop the fully functional database with User Interface and appropriate outputs. Populate the database with appropriate test data and fully test the system using appropriate tests, such as security, referential integrity, and functionality and use the testing outcomes to optimize it.**

## **Introduction:**

The conceptual foundation of relational databases is the relational model. It is a method for organizing data that involves the use of a language, structures, and relationships that are represented by a set of interconnected tables.

In this task, I have produced a design for a relation database management to meet clients requirements. Diagrams such as Entity Relationship (ER) diagram, Data Flow Diagram (DFD), Class diagram and Database Schema has been presented to overview the workflow of the designed system. Proper analyzation of how the design will optimize system performance has been carried out. Using C# as the back end software application and Microsoft SQL server as database , a fully functional system to meet the clients and system requirements has been developed and the system was tested for functionality and performance. Moreover, I have implemented the effective features in the solution to handle concurrency, security, user authorizations and data recovery and have critically evaluated the system.

## **Entity Relationship Diagram:**

Entity Relationship Diagram, also called Entity Relationship Model is a visual representation of different data using conventions that illustrates how entities relate to each other within a system (Anon., n.d.). It is a graphical representation used in computing typically for organization of data. ER diagram are frequently used during development stage process to identify different elements and their relationships with each other. In other words, The relationships between entity sets recorded in a database are depicted in an entity relationship diagram (ERD). An object, or a data component, is an entity in this context. A collection of similar entities is referred to as an entity set. These entities can have qualities defined via attributes. The various elements that are involved in ER diagram.

- **Entity:** An entity is an object or component of data. An entity is represented as rectangle in an ER diagram. An entity is a definable thing, which can be a person, place, event, or object within a system. In Library management system the entities are Books, Category, Fines, Issue Book, Library Card, Reservation, Login, Users
- **Attribute:** An attribute a property, trait, or characteristic of entity, relationship or another attribute (Anon., n.d.). In Library management system the attributes are:  
 Book: BookID, BookName, BookCategoryID, AuthorID and SubjectID  
 Subjects: SubjectID, SubjectName, Remarks  
 BookCategory: BookCategoryID, BookCategoryName and Description  
 AssesionMapping: AssesionID, BookID, Status  
 Author: AuthorID, AuthorName, and Description

BookIssueReturn: BookIssueReturnID, MemberID, AssesionID, IssueDate, DueDate, ReturnDate: FineAmount, Status, StaffMemberID, Remarks  
 Member: MemberID, MemberName, StaffMemberID, MemberAddress, MemberCategory: StaffMemberID, MemberCategoryName, Remarks  
 Fine: FineID, LateDate, StaffMemberID, Amount, Remarks

### **Relationships between Entities:**

A relationship shows how different entities are related with each other. There are four types of relationship between entities:

- **One-to -one:** The relationship can be one-to-one when only one instance of an entity is associated with other instance of another entity. In Library management system, a User has one particular roles.
- **One-to-Many:** The relationship can be one-to-many when one instance of an entity is associated with one or many instances of another entity. In Library management system, a User has one or more books.
- **Many-to-One:** The relationship can be many-to-one when many instances of an entity is associated with one instance of another entity. In Library management system, many Users are associated with one single books.
- **Many-to-Many:** The relationship can be many-to-many when one instance of an entity A is associated with one, zero or many instances of entity B and another instance of entity B is associated with one, zero or many instances of entity A. In Library management system, a publishers have many books.

### **ER Diagram Symbols:**

In ER Diagram, several symbols are used to represent the information of the system. Different symbols different elements:

- Rectangular Boxes represents entities.
- Circle represents attributes.
- Diamond represents relationships.

## ER Diagram for Library Management System

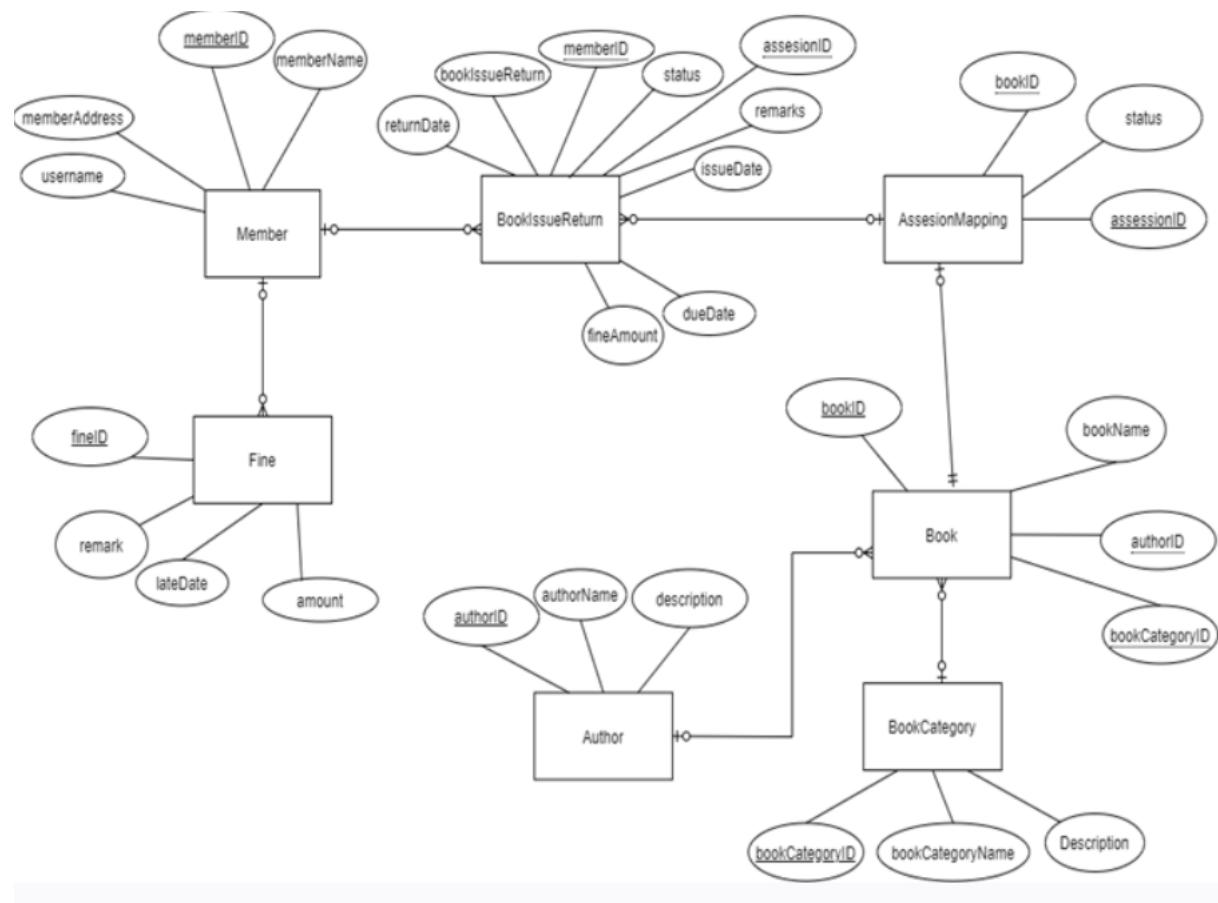


Figure: ER Diagram for Library Management System

### DFD (Data Flow Diagram):

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow. **(Lucidchart, 2018)** A DFD, like the best diagrams and charts, can frequently graphically "express" things that are difficult to describe in words, and it can be used by both technical and nontechnical audiences, from developers to CEOs. That is why DFDs have remained so popular over the years. While they are useful for displaying data flow software and systems, they are less useful for interactive, real-time, or database-oriented software and systems nowadays

### Components of DFDs:

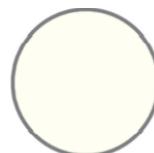
DFDs consist of four major components that have particular symbols. Those symbols are used to show the components interact with each other. The components are described below:

- **External Entity:** External entities are any object like person, hardware or another software who/which either consume data from the software or provide data to the software. They are known as source or destination of the data and are drawn on the edges of diagram. External entity is represented as rectangle.



*Figure: External Entity*

- **Processes:** A process is a function or a piece of code, which is used for manipulation, and transformation of data. Processes receive input and generates output. Process is represented by circle or bubbles.



*Figure: Process*

- **Data Flows:** Data flow is the movement of data between entity, the process and data store that portrays the nature of the data used. It is represented by arrow labeled with short data name.



*Figure: Data Flow*

- **Data Store:** Data store is where data produced by the process or required for future use are stored. Files and tables can be considered as data store. It is represented by a set of parallel lines.

Data store

*Figure: Data Store*

### Advantages and Disadvantages of DFDs:

The advantages of DFDs are as follows:

- DFDs gives information about data as well as functionality to software designer.
- It provides complete overview of system components.
- DFDs are easier to understand by technical and non-technical user.
- Structure of DFD allows starting from a broad overview and expand it to a hierarchy of detailed diagrams.

### The disadvantages of DFDs are as follows:

- DFDs for large system can become difficult to translate and read.
- DFDs are time consuming.
- DFDs are useless without prerequisites details and become confusing to programmers sometimes.

**1. Zero level DFD:** DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers. (**Lucidchart, 2018**)

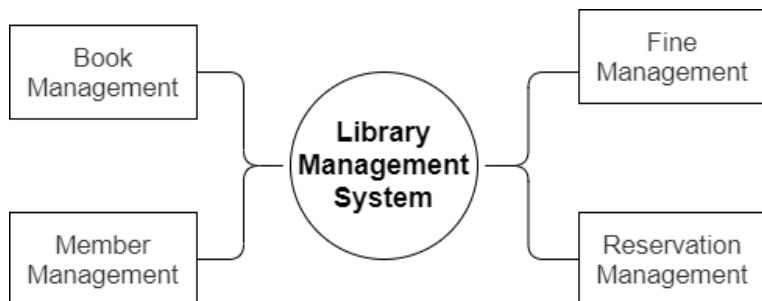


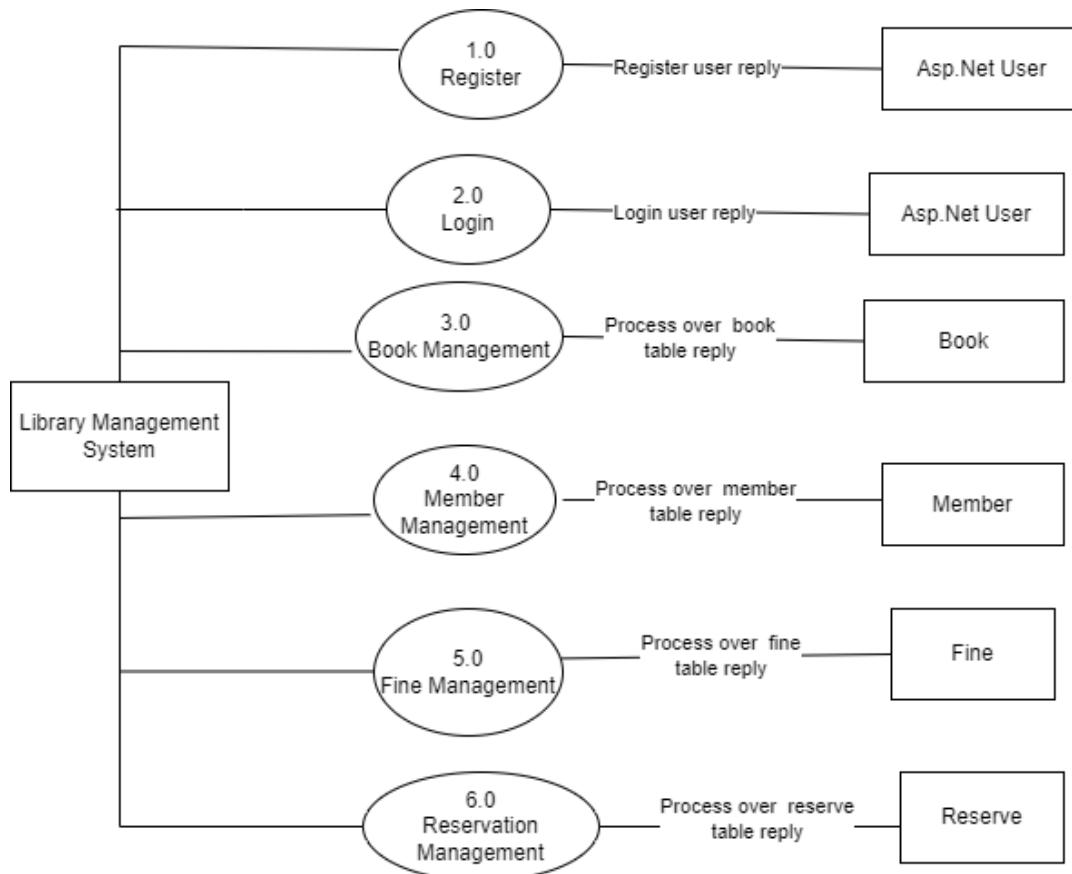
Figure: Zero level DFD

### 2. Level one DFD:

The Level 0 DFD is broken down into more specific, Level 1 DFD. Level 1 DFD depicts basic modules in the system and flow of data among various modules. Level 1 DFD also mentions basic processes and sources of information.

- It provides a more detailed view of the Context Level Diagram.

- Here, the main functions carried out by the system are highlighted as we break into its sub-processes.



*Figure:Level one DFD*

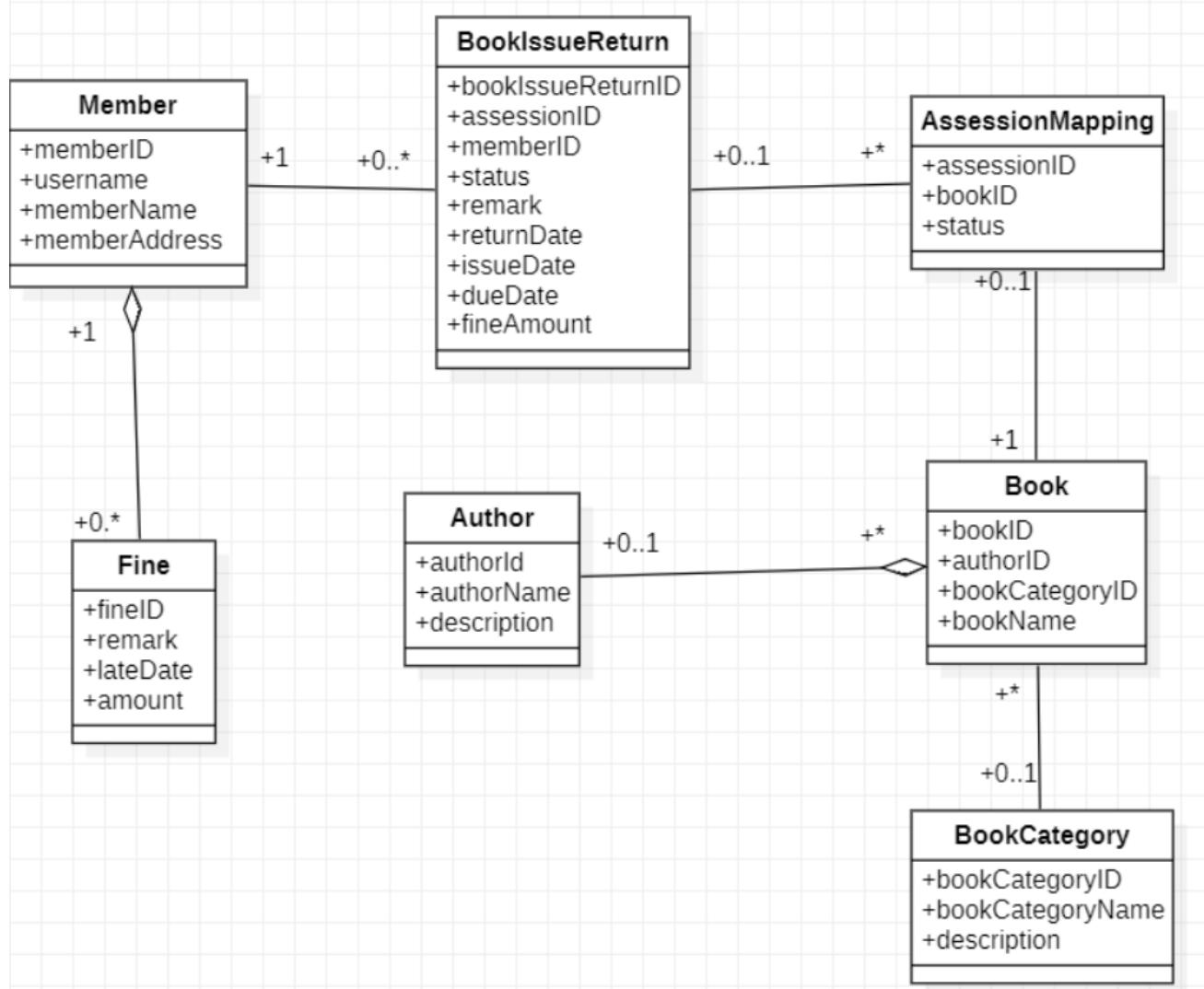
### Database Schema

"A database schema is a blueprint for the database's structure." The schema instructs the database engine, or the computer that runs the database, on how to insert and retrieve data. The schema serves as a foundation for database operations and content. It's a set of metadata that explains the relationships between items and data in a database" (**GeeksforGeeks, 2020**) It can be thought of as a box that contains tables, stored procedures, views, and other data assets. This box's infrastructure is defined by a schema. The database schema is a design, or a list of properties and instructions, that instructs the database engine how the data is arranged and how the various components are connected. A database schema typically shows the different tables, their fields and the relationship between them and other tables. The database schema is defined within the native database language; therefore the logical structure and visualization of schema might vary in each database language. It helps database administrators in understanding the architectural layout of the database.

No.	Class	Attributes
1.	Book	BookID, BookName, BookCategoryID, AuthorID and SubjectID
2.	Subjects	SubjectID, SubjectName, Remarks
3.	BookCategory	BookCategoryID, BookCategoryName and Description
4.	AssesionMapping	AssesionID, BookID, Status
5.	Author	AuthorID, AuthorName, and Description
6.	BookIssueReturn	BookIssueReturnID, MemberID, AssesionID, IssueDate, DueDate, ReturnDate, FineAmount, Status, StaffMemberID, Remarks
7.	Member	MemberID, MemberName, StaffMemberID, MemberAddress,
8.	MemberCategory	StaffMemberID, MemberCategoryName, Remarks
9.	Fine	FineID, LateDate, StaffMemberID, Amount, Remarks

## Class Diagram

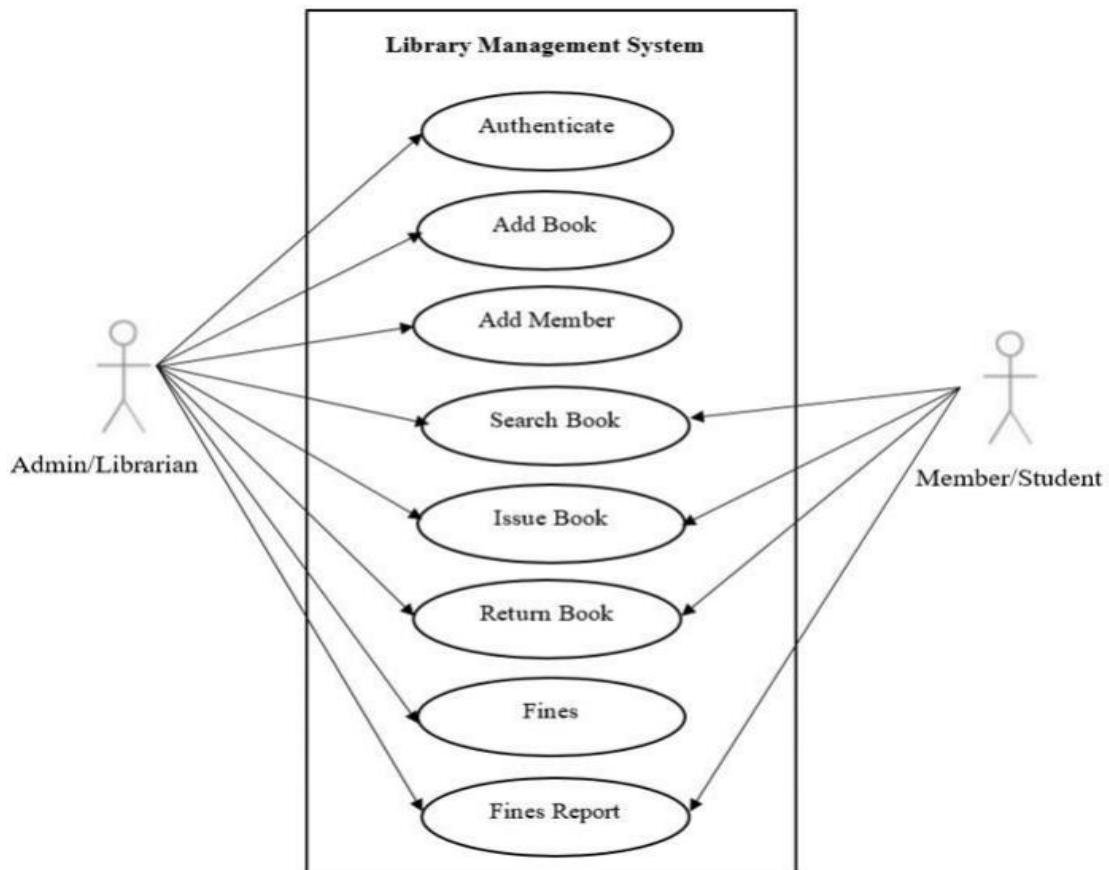
"A class diagram is a form of static structure diagram used in software engineering to depict the structure of a system by displaying the system's classes, properties, actions (or methods), and relationships among objects." It depicts an application's static view. Class diagrams are utilized not only for visualizing, describing, and documenting many parts of a system, but also for creating executable code for software applications." (**GeeksforGeeks, 2020**) Because they are the only type of UML diagram that can be translated directly with object oriented languages, they are frequently utilized in software development in object oriented systems modeling. Class diagrams, also known as structural diagrams, illustrate a collection of classes, interfaces, connections, collaborations, and constraints. The major goal of a class diagram is to describe a static view of an application. They are the only diagrams that can be directly mapped with object-oriented languages, making them popular during development. Unlike the activity diagram and sequence diagram, which can only show the application's sequence flow, the class diagram is the most popular UML diagram in our coder community.

*Figure: Class Diagram*

### Use Case Diagram

A use case diagram is a way to summarize details of a system and the users within that system. It is generally shown as a graphic depiction of interactions among different elements in a system. Use case diagrams will specify the events in a system and how those events flow, however, use case diagram does not describe how those events are implemented.

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.



*Figure: Use case diagram*

### Analyzing design of relational database for optimization of system performance

The relational database's design allows us to model information and data as a series of tables with rows and columns. Each column of the relation represents a data attribute, and each row represents a record. I created diagrams such as entity relationships, data flow diagrams, database schemas, and class diagrams for the activities listed above. Those pictorial representations aren't just for programmers; they're also for business folks and project managers since they make the system's work flow and representation easy to understand.

The ACID (atomicity, consistency, integrity, and durability) attributes of a database design are met by the relational database design. These designs also need the usage of a database server in applications to cope with data management issues, which eventually aids in system performance optimization. Defining relations, primary keys, defining relationships, and finally implementing normalization are the four stages of relational database design. Using relational database design in analysis and design iteratively, we may establish understanding between the business group and the development group regarding the requirement and the process that has to occur in the system to match our client expectations. During the various stages of software development, there is always the possibility of modifying the business requirements. In these circumstances, we can re-design, code, and test the system by investing some time and money.

In a variety of ways, the design of a relational database aids system performance. To begin with, design aids in specifying the structure and system behavior, i.e., we may learn about the application's workflow and how it will behave when the development phase is completed. It also aids in the visualization of the system, allowing us to learn more about the system's proposed features and how they will be implemented. The most important thing that design does for us is providing a template that guides us through the process of building a system. We can figure out what features we need to incorporate when we create the data flow diagram and entity relationship diagram. These designs will benefit not just the program, but also other businesspeople who are unfamiliar with software development. The design of a data flow diagram helps to describe the system's boundaries. It is useful for informing users about existing system knowledge. These diagrams are more easily understood by both technical and non-technical audiences and can provide a high-level overview of the system. They aid system designers and others in visualizing the current system or one that may be required to satisfy new needs during the initial analysis stages. Data flow diagrams are preferred by system designers when a clear knowledge of the boundary between current and prospective systems is required.

Let's look at how the design of a class diagram can aid with system performance. Implementing the class diagram during the development phase of our application forces the programmer to consider the structure of classes and how they will interact before writing any code. This could result in a more stable application. The class diagram also serves as a blueprint for maintenance programmers, allowing them to obtain a sense of how the application is organized before diving into the code. This could cut down on maintenance time. As a result, class diagram aids in the optimization of system performance by assisting us in the creation of more resilient applications and saving maintenance time. The entity relationship diagram is used to present the relation of various tables present in a database. Having an

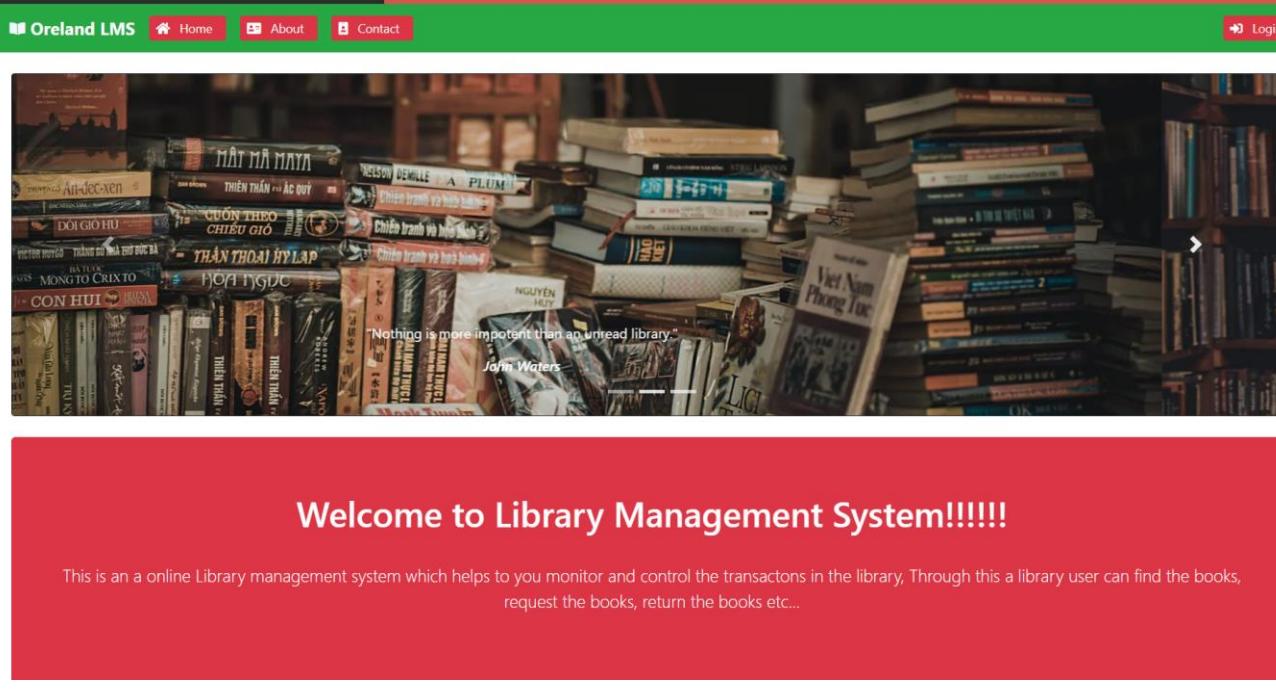
effective design provides assistance to the database designers to determine the flow of data and working of the complete system. Entity relationship diagrams are an essential part of the business organizations as they prove to be beneficial in managing wide data in an easy and effective manner. It acts as a blueprint of the existing database and allows the designers to create an accurate design as per the needs and requirements of the company and the project. The ERD model makes data flow more efficient. These creative and simple diagrams serve as the best tool for the business organizations allowing them to maintain their database effectively.

In this approach, all of those designs assist us in specifying the structure and system behavior, i.e., we can learn about the application's workflow and how it will behave when the development process is through. It also aids in the visualization of the system, allowing us to learn more about the system's proposed features and how they will be implemented. As a result, I would also advise using relational model design in software development to improve system speed.

### **Design of Library Management System**

The Library management system for Oreland college is a web application which is going to be developed using C# .NET framework. As our client requirement is of application the design of application is done using the web forms. The database used for this application is MS SQL where I have successfully designed the database as shown in previous tasks. For the successful completion of project I am going to use Visual Studio as my IDE as it is one of the best development tool and most of the .Net developers use it to develop .net applications. Below I have provided the design of Library management system forms:

**1. Home Page:** This Is the home page of Oreland Library where we get access of different things



Welcome to Library Management System!!!!!!

This is an online Library management system which helps to you monitor and control the transactions in the library. Through this a library user can find the books, request the books, return the books etc...

```

1   <!DOCTYPE html>
2   <html>
3   <head>
4       <meta charset="utf-8" />
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>@ViewBag.Title - UrLMS</title>
7       <link href("~/Content/Site.css" rel="stylesheet" type="text/css" />
8       <link href("~/Content/bootstrap.min.css" rel="stylesheet" type="text/css" />
9       <link href("~/Content/Fontawesome-all.css" rel="stylesheet" />
10      <link href("~/Content/lmscustom.css" rel="stylesheet" />
11      <script src "~/Scripts/modernizr-2.8.3.js"></script>
12  </head>
13  <body style="padding:unset">
14  <div>
15      <div style="background-color:#5cb85c;">
16          <div style="background-color:#d9534f; height:8px; width:450px">
17          </div>
18      </div>
19      <nav class="navbar navbar-expand-lg navbar-dark bg-success">
20          <span class="icon">
21              <i class="fas fa-book-open"></i>
22          </span>
23          @Html.ActionLink("Oreland LMS", "Index", "tblBooks", new { area = "" }, new { @class = "navbar-brand font-weight-bold" })
24          <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"
25              aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
26              <span class="navbar-toggler-icon"></span>
27          </button>
28          <div class="collapse navbar-collapse" id="navbarSupportedContent">
29              <ul class="navbar-nav mr-auto">
30                  <li class="nav-item btn-sm btn-danger">
31                      <i class="fas fa-home"></i>

```

**2. About:** This is the About page where we can get information about the Oreland Library

**Bussiness Information**

- Opened on 14 Augest 2018
- Mission

Our vision is to provide modern library management system.

**Meet Our Team**

- Anil Bhusal
- Abik Mushyakho
- Bikesh Bhaila
- Saurav Lamichhane
- Nabin Basnet
- Rekha Shrestha

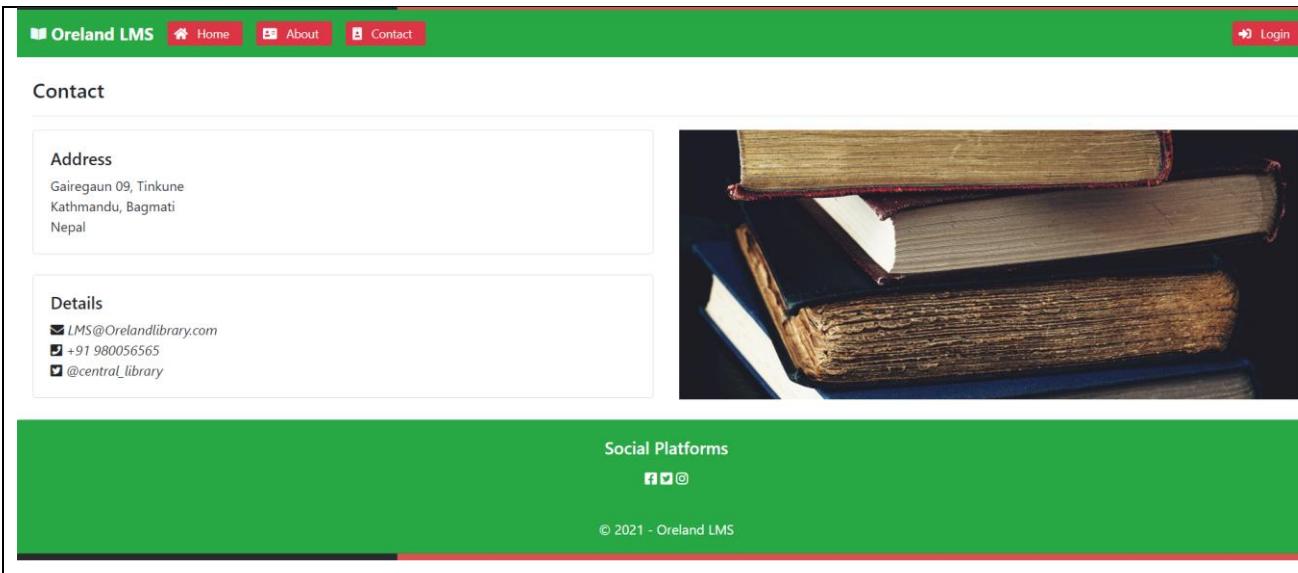
**Social Platforms**

© 2021 - Oreland LMS

```

1  @{
2      ViewBag.Title = "About";
3      Layout = "~/Views/Shared/_MainLayout.cshtml";
4  }
5
6
7      <br />
8      <h4>About</h4>
9      <hr />
10
11     <div class="row">
12         <div class="col-6">
13             <div class="card card-body">
14                 <h5>Bussiness Information</h5>
15                 <div>
16                     <i class="fas fa-flag"></i>
17                     <span>Opened on 14 Augest 2018</span>
18                 </div>
19                 <div>
20                     <i class="fas fa-brain"></i>
21                     <span>Mission</span>
22                     <p>Our vision is to provide modern library management system.</p>
23                 </div>
24             </div>
25             <br />
26             <div class="card card-body">
27                 <h5>Meet Our Team</h5>
28                 <div>
29                     <i class="fas fa-user-cog"></i>
```

3. Contact : This is the contact page of LMS of Oreland

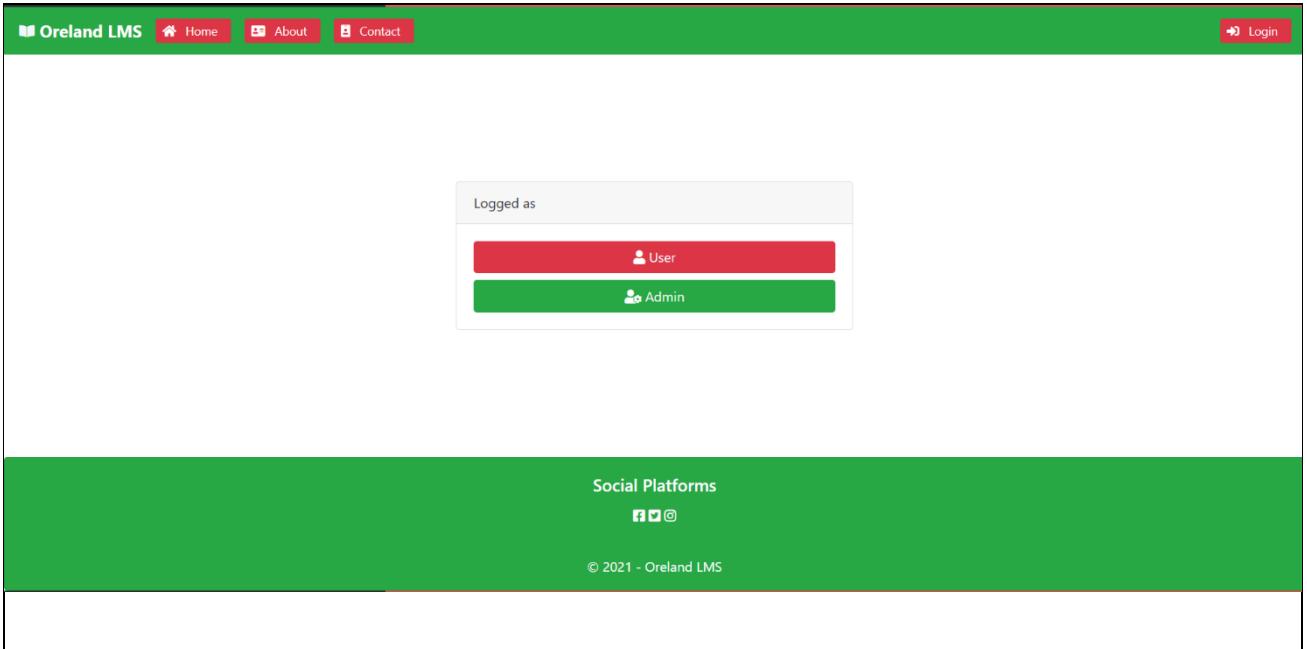


The screenshot shows the Contact page of the Oreland LMS. It features a green header bar with the logo and navigation links for Home, About, Contact, and Login. The main content area has two cards: one for Address (Gairegaun 09, Tinkune, Kathmandu, Bagmati, Nepal) and one for Details (email: LMS@Orelandlibrary.com, phone: +91 980056565, and a social media handle). To the right is a stack of old books. Below the cards is a green banner with Social Platforms (Facebook, Twitter, Instagram) and a copyright notice: © 2021 - Oreland LMS.

```

1  @{
2      ViewBag.Title = "Contact";
3      Layout = "~/Views/Shared/_MainLayout.cshtml";
4  }
5
6
7      <br />
8      <h4>Contact</h4>
9      <hr />
10
11     <div class="row">
12         <div class="col-6">
13             <div class="card card-body">
14                 <h5>Address</h5>
15                 <div>
16                     <span>Gairegaun 09, Tinkune</span>
17                 </div>
18                 <div>
19                     <span>Kathmandu, Bagmati</span>
20                 </div>
21                 <div>
22                     <span>Nepal</span>
23                 </div>
24
25
26             </div>
27             <br />
28             <div class="card card-body">
29                 <h5>Details</h5>
30                 <div>
31                     <i class="fas fa-envelope"></i>
32                 </div>
33             </div>
34         </div>
35     </div>
36 
```

**4. Login Form:** The login form is the preliminary form of the application. With the use of login form only, the authenticate users can log into the library management system. Whenever, a login form is displayed, the authenticate user can log into the system filling the appropriate details. In case of Library management system, the users have to provide genuine Username and Password. After providing the valid information, the user can enter in to the system clicking the login button. After proper validation, if the entered Username and Password match with the data stored in the database, the user is successfully logged into the system and redirected to the dashboard form of the application as per the respective user type.



The screenshot shows the Oreland LMS login page. At the top, there's a green header bar with the logo 'Oreland LMS' and navigation links for 'Home', 'About', and 'Contact'. On the right side of the header is a 'Login' button. Below the header, a modal window titled 'Logged as' is displayed. It contains two buttons: a red one labeled 'User' with a user icon, and a green one labeled 'Admin' with an admin icon. The main content area below the modal has a green background and features a 'Social Platforms' section with icons for Facebook, Twitter, and LinkedIn. At the bottom of the page, there's a copyright notice: '© 2021 - Oreland LMS'.

```

@{
    ViewBag.Title = "Login";
    Layout = "~/Views/Shared/_MainLayout.cshtml";
}

<div style="padding-bottom:10%; padding-top:10%" class="col-4 offset-4">
    <div class="card">
        <div class="card-header">Logged as</div>
        <div class="card-body">
            <div class="btn btn-block btn-danger" data-toggle="tooltip" data-placement="right" title="Logged as user">
                <i class="fas fa-user"></i>
                @Html.ActionLink("User", "Login", "User", null, new { style = "text-decoration:none", @class = "text-light" })
            </div>
            <div class="btn btn-block btn-success" data-toggle="tooltip" data-placement="right" title="Logged as admin">
                <i class="fas fa-user-cog"></i>
                @Html.ActionLink("Admin", "Login", "Admin", null, new { style = "text-decoration:none", @class = "text-light" })
            </div>
        </div>
    </div>
</div>

```

**For User Login:**

Login

UserEmail

saurav@gmail.com

UserPass

.....

**For admin:** In this form, admin can access by using email and password. If email or password is incorrect than it show the message as user name or password are not matching.

Login

Email address

saurav@ismt.edu

Password

.....

**Admin Panel:** Here in this admin panel, admin can access the form they need. After login the admin panel is open. This main page is the page where there is a record of all book list.

The screenshot shows a web application interface for the Oreland LMS. At the top, there is a navigation bar with links for Home, About, Contact, Books, Users, Request, Accepted, Return, Welcome back Admin!! (logged in), and Logout. Below the navigation bar is a section titled "Book list" with a table displaying three entries:

Title	Category	Author	Copies	Publication	Pub Name	ISBN	Copyright	Date Added	Status	Actions
Muna Madan	Romance	Kali Baskota	18	Asmita	Asmita Publication	0-892-90282-1	2018	2021-01-20	New	<button>Edit</button> <button>Details</button> <button>Delete</button>
Subtle Art	Action	Jhoeen	10	Syplex	Hunted	0-123-12345-1	2012	11/21/2018	Old	<button>Edit</button> <button>Details</button> <button>Delete</button>
Watch Me	Fiction	Roberto	0	Dunlep	Asmita	0-123-13245-1	2017	11/08/2018	Old	<button>Edit</button> <button>Details</button> <button>Delete</button>

Below the table, it says "Showing 1 to 3 of 3 entries". To the right, there are buttons for Previous, Next, and a page number (1). At the bottom of the page, there is a "Social Platforms" section with icons for Facebook, Twitter, and Instagram, and a copyright notice: "© 2021 - Oreland LMS".

The bottom half of the screenshot shows the source code for the navbar partial view. The code uses Bootstrap classes and ASP.NET MVC helpers to generate the navigation menu. It includes links for Home, About, Contact, and various administrative actions like Edit, Details, and Delete.

```

19 <nav class="navbar navbar-expand-lg navbar-dark bg-success">
20   <span class="icon">
21     <i class="fas fa-book-open"></i>
22   </span>
23   @Html.ActionLink("Oreland LMS", "Index", "tblBooks", new { area = "" }, new { @class = "navbar-brand font-weight-bold" })
24   <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"
25     aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
26     <span class="navbar-toggler-icon"></span>
27   </button>
28   <div class="collapse navbar-collapse" id="navbarSupportedContent">
29     <ul class="navbar-nav mr-auto">
30       <li class="nav-item btn-sm btn-danger">
31         <i class="fas fa-home"></i>
32         @Html.ActionLink("Home", "AdminHome", "AdminTransaction", new { area = "" }, new { @class = "nav-link text-decoration-none" })
33       </li>
34       <li class="nav-item btn-sm btn-danger mx-3">
35         <i class="fas fa-address-card"></i>
36         @Html.ActionLink("About", "AdminAbout", "AdminTransaction", new { area = "" }, new { @class = "nav-link text-decoration-none" })
37       </li>
38       <li class="nav-item btn-sm btn-danger">
39         <i class="fas fa-id-badge"></i>
40         @Html.ActionLink("Contact", "AdminContact", "AdminTransaction", new { area = "" }, new { @class = "nav-link text-decoration-none" })
41       </li>
42     </ul>
43     <ul class="navbar-nav justify-content-end" style="padding-right:50px">
44       <li class="" style="padding-left:5px; padding-right:5px">
45         @Html.ActionLink("Books", "Index", "TblBooks", new { area = "" }, new { @class = "nav-item btn-sm btn-primary" })
46       </li>

```

**Add Book:** This form is used to add the record of the book title, category, author, publication name, date, etc. Button is used to submit the added book. To add the new book we have to give the book records.

**Add book**

---

Title <input type="text" value="Catch Me If You Can"/>	Category <input type="text" value="Select catagory"/>
Author <input type="text" value="Frank Abagnale Jr"/>	Copies <input type="text" value="15"/>
Publication <input type="text" value="Grosset &amp; Dunlap"/>	Pub Name <input type="text" value="Hunted Forger"/>
ISBN <input type="text" value="0-448-16538-4"/>	Copyright <input type="text" value="1980"/>
Date Added <input type="text" value="05/23/1983"/>	Status <input type="text" value="Select status"/>
<input type="button" value="Submit"/>	

[← Back](#)

---

**Book List:** After adding the book, we can see the list on the book list.

**Book list**

---

Title	Category	Author	Copies	Publication	Pub Name	ISBN	Copyright	Date Added	Status	Actions
Muna Madan	Romance	Kali Baskota	18	Asmita	Asmita Publication	0-892-90282-1	2018	2021-01-20	New	<input checked="" type="checkbox"/> Edit <input type="button" value="Details"/> <input type="button" value="Delete"/>
Subtle Art	Action	Jhoeen	10	Syplex	Hunted	0-123-12345-1	2012	11/21/2018	Old	<input checked="" type="checkbox"/> Edit <input type="button" value="Details"/> <input type="button" value="Delete"/>
Watch Me	Fiction	Roberto	0	Dunlep	Asmita	0-123-13245-1	2017	11/08/2018	Old	<input checked="" type="checkbox"/> Edit <input type="button" value="Details"/> <input type="button" value="Delete"/>

Showing 1 to 3 of 3 entries

[Previous](#) 1 [Next](#)

**Social Platforms**

© 2021 - Oreland LMS

**Book Edit:** In book edit we can click on any box where is needed to be update. We can update all the book details. Button is used to update the information of the book.

Saurav lamichhane (HND/ Third Semester Section “P5”)

43

**Update book**

Title <input type="text" value="Muna Madan"/>	Category <input type="text" value="Romance"/>
Author <input type="text" value="Kali Baskota"/>	Copies <input type="text" value="18"/>
Publication <input type="text" value="Asmita"/>	Pub Name <input type="text" value="Asmita Publication"/>
ISBN <input type="text" value="0-892-90282-1"/>	Copyright <input type="text" value="2018"/>
Date Added <input type="text" value="2021-01-20"/>	Status <input type="text" value="New"/>
<input type="button" value="Update"/>	

[← Back](#)

**Book Details:** Here in this form, the title, category, author, publication, etc. are given that are asked by user. Button is used for to edit the book details.

**Book details**

Title Muna Madan	Category Romance
Author Kali Baskota	Copies 18
Publication Asmita	Pub Name Asmita Publication
ISBN 0-892-90282-1	Copyright 2018
Date Added 2021-01-20	Status New
<input type="button" value="Edit"/>	

[← Back](#)

**Book Delete:** This form is to delete the book information. We can delete all the information about the book in this page. To delete the information we have to give the title, category, publication, author, etc.

**Oreland LMS** [Home](#) [About](#) [Contact](#)

[Books](#) [Users](#) [Request](#) [Accepted](#) [Return](#)

Welcome back Admin!! [Logout](#)

Are you sure you want to delete this?

Title Muna Madan	Category Romance
Author Kali Baskota	Copies 18
Publication Asmita	Pub Name Asmita Publication
ISBN 0-892-90282-1	Copyright 2018
Date Added 2021-01-20	Status New

[Delete](#)

[← Back](#)

**User List:** This form is to show the information of the user. Like name, gender, department, email address, etc. are listed

**Oreland LMS** [Home](#) [About](#) [Contact](#)

[Books](#) [Users](#) [Request](#) [Accepted](#) [Return](#)

Welcome back Admin!! [Logout](#)

User list

[Add](#)

UserName	UserGender	UserDep	UserAdmNo	UserEmail	UserPass	Actions
abik	Male	Ec	2017912	abik.mushyakho94@gmail.com	P@ssw0rd	<a href="#"><span style="color: black;">Edit</span></a> <a href="#"><span style="color: yellow;">Details</span></a> <a href="#"><span style="color: red;">Delete</span></a>
bikeshbhaila	Male	CS	2015350	bikesh@gmail.com	bikesh@123	<a href="#"><span style="color: black;">Edit</span></a> <a href="#"><span style="color: yellow;">Details</span></a> <a href="#"><span style="color: red;">Delete</span></a>
Nabin	Male	Civil	20170612	nabin@gmail.com	nabin@123	<a href="#"><span style="color: black;">Edit</span></a> <a href="#"><span style="color: yellow;">Details</span></a> <a href="#"><span style="color: red;">Delete</span></a>
rekha11	Male	Ec	5	rekha11@gmail.com	nabin123@	<a href="#"><span style="color: black;">Edit</span></a> <a href="#"><span style="color: yellow;">Details</span></a> <a href="#"><span style="color: red;">Delete</span></a>
Saurav	Male	Mech	2016309	saurav@gmail.com	saurav@123	<a href="#"><span style="color: black;">Edit</span></a> <a href="#"><span style="color: yellow;">Details</span></a> <a href="#"><span style="color: red;">Delete</span></a>

Show 10 entries Search:

Showing 1 to 5 of 5 entries Previous [1](#) Next

Social Platforms [Facebook](#) [Twitter](#) [Instagram](#)

**User Add:** This form is to add the user. User have to give their name, gender, email address, etc. information to the admin. Button is used for to submit the information.

**Add user**

---

UserName <input type="text" value="Soubhagya P"/>	UserGender <input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other
UserDep <input type="text" value="Select department"/>	UserAdmNo <input type="text" value="2015360"/>
UserEmail <input type="text" value="sou@gmail.com"/>	UserPass <input type="text" value="123123pwd"/>
<input type="button" value="Submit"/>	

[← Back](#)

**Social Platforms**

© 2021 - Oreland LMS

**User Edit** In this form user can update their given information like name, gender, department, email address, etc

**Update user**

---

UserName <input type="text" value="abhi&amp;"/>	UserGender <input checked="" type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other
UserDep <input type="text" value="It"/>	UserAdmNo <input type="text" value="2017912"/>
UserEmail <input type="text" value="abik.mushyakho94@gmail.com"/>	UserPass <input type="text" value="P@ssw0rd"/>
<input type="button" value="Update"/>	

[← Back](#)

**Social Platforms**

© 2021 - Oreland LMS

**User Details:** This form is to show the added details of the user. They can also edit their information in this form.

The screenshot shows a 'Book details' section with the following data:

Title	Muna Madan	Category	Romance
Author	Kali Baskota	Copies	18
Publication	Asmita	Pub Name	Asmita Publication
ISBN	0-892-90282-1	Copyright	2018
Date Added	2021-01-20	Status	New

Edit

[← Back](#)

**User Delete:** This form is to delete the information of the user. To delete the information of the user we have to give user name, gender, department, admission number, etc

The screenshot shows a confirmation message: "Are you sure you want to delete this?" followed by the same book details table as the previous screenshot.

Delete

[← Back](#)

### See Book Request List

This form is to show the book request list of the user. User have to give the book title and username for the requested book.

**Oreland LMS** [Home](#) [About](#) [Contact](#)

[Books](#) [Users](#) [Request](#) [Accepted](#) [Return](#)      Welcome back Admin!! [Logout](#)

**Request list**

Show 10 entries Search:

Book Title	Book ISBN	Request Date	User Name	Action
Catch Me	0-123-12345-1	11/21/2021	Saurav	<a href="#">Accept</a> <a href="#">Reject</a>

Showing 1 to 1 of 1 entries

**Social Platforms**  
Facebook Twitter Instagram

© 2021 - Oreland LMS

**See Accepted Request:** This form is to show the list of the accepted book. After the request of the book by the user admin have to accept the request and it show the accepted book lists.

**Oreland LMS** [Home](#) [About](#) [Contact](#)

[Books](#) [Users](#) [Request](#) [Accepted](#) [Return](#)      Welcome back Admin!! [Logout](#)

**Accepted list**

Show 10 entries Search:

Book Title	Book ISBN	Accepted Date	User Name
Catch Me	0-123-12345-1	11/10/2021	Saurav
Muna Madan	0-892-90282-1	11/10/2021	bikeshbhaila
Muna Madan	0-892-90282-1	11/10/2021	Saurav
Watch Me	0-123-13245-1	11/10/2021	Nabin
Watch Me	0-123-13245-1	11/20/2021	abik

Showing 1 to 5 of 5 entries

**Social Platforms**  
Facebook Twitter Instagram

© 2021 - Oreland LMS

**See Return Book:** List This form is to show the list of the return book by the user. Any book that is return by the user can see the list in this page.

**Return list**

Show 10 entries Search:

Book Title	Book ISBN	Return Date	User Name	Action
Muna Madan	0-892-90282-1	24-Nov-21	bikeshbhaila	<button>Accept</button>

Showing 1 to 1 of 1 entries Previous 1 Next

Thank you for your work

Social Platforms [Facebook](#) [Twitter](#) [Instagram](#)

**For User:** This form is to show the book list for the user only.

Oreland LMS [Home](#) [About](#) [Contact](#) [Borrow](#) [Requested](#) [Received](#) [Rejected](#) Welcome back Saurav!! [Logout](#)

**Book list**

Show 10 entries Search:

Title	Category	Author	Copies	Publication	Pub Name	ISBN	Copyright	Date Added	Status	Actions
Muna Madan	Romance	Kali Baskota	18	Asmita	Asmita Publication	0-892-90282-1	2018	2021-01-20	New	<button>Request</button>
Subtle Art	Action	Jhoeen	10	Syplex	Hunted	0-123-12345-1	2012	11/21/2018	Old	<button>Request</button>
Watch Me	Fiction	Roberto	0	Dunlep	Asmita	0-123-13245-1	2017	11/08/2018	Old	<button>Request</button>

Showing 1 to 3 of 3 entries Previous 1 Next

Social Platforms [Facebook](#) [Twitter](#) [Instagram](#)

© 2021 - Oreland LMS

**Requested list:** This for is to show the requested book list of the user. Like date and ISBN.

Oreland LMS [Home](#) [About](#) [Contact](#) [Borrow](#) [Requested](#) [Received](#) [Rejected](#) Welcome back Saurav!! [Logout](#)

**Requested list**

Catch Me

Book ISBN  
0-123-12345-1  
Requested Date  
11/21/2021

[Cancel](#)

Social Platforms [Facebook](#) [Twitter](#) [Instagram](#)

© 2021 - Oreland LMS

**Received List:** This form is to show the received list of the book by the user.

The screenshot shows the 'Received list' section of the Oreland LMS. It displays two entries, each with a thumbnail, title, book ISBN, received date, and a 'Return' button.

Book Title	Book ISBN	Received Date	Action
Muna Madan	0-892-90282-1	11/10/2021	<a href="#">Return</a>
Catch Me	0-123-12345-1	11/10/2021	<a href="#">Return</a>

**Social Platforms**  
[Facebook](#) [Twitter](#) [Instagram](#)  
 © 2021 - Oreland LMS

**Rejected List:** This form is to show the rejected list of the book.

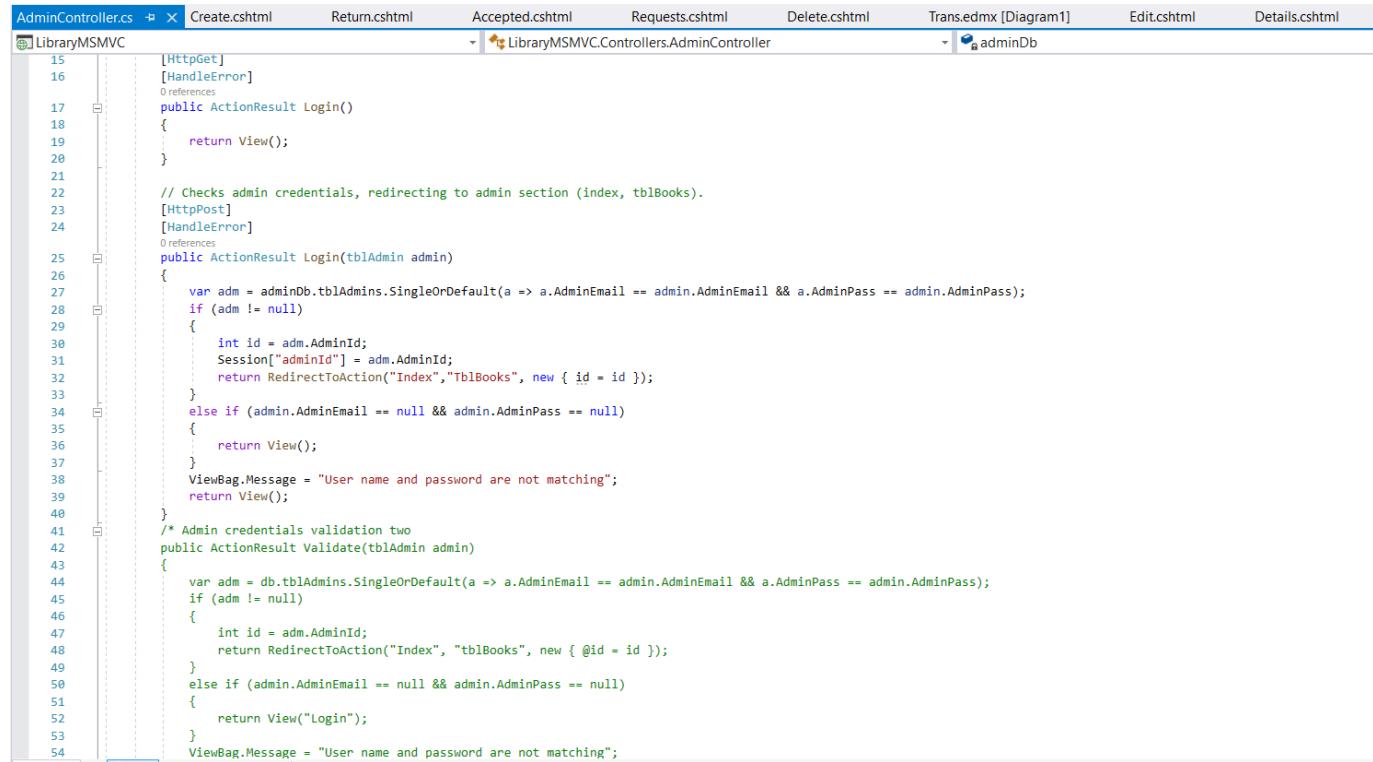
The screenshot shows the 'Rejected list' section of the Oreland LMS. It displays one entry with a thumbnail, title, book ISBN, rejected date, and two buttons: 'Rerequest' and 'Cancel'.

Book Title	Book ISBN	Rejected Date	Action
Catch Me	0-123-12345-1	12/8/2021	<a href="#">Rerequest</a> <a href="#">Cancel</a>

**Social Platforms**  
[Facebook](#) [Twitter](#) [Instagram](#)  
 © 2021 - Oreland LMS

## Some important Controller codes of LMS application

### Admin controller:

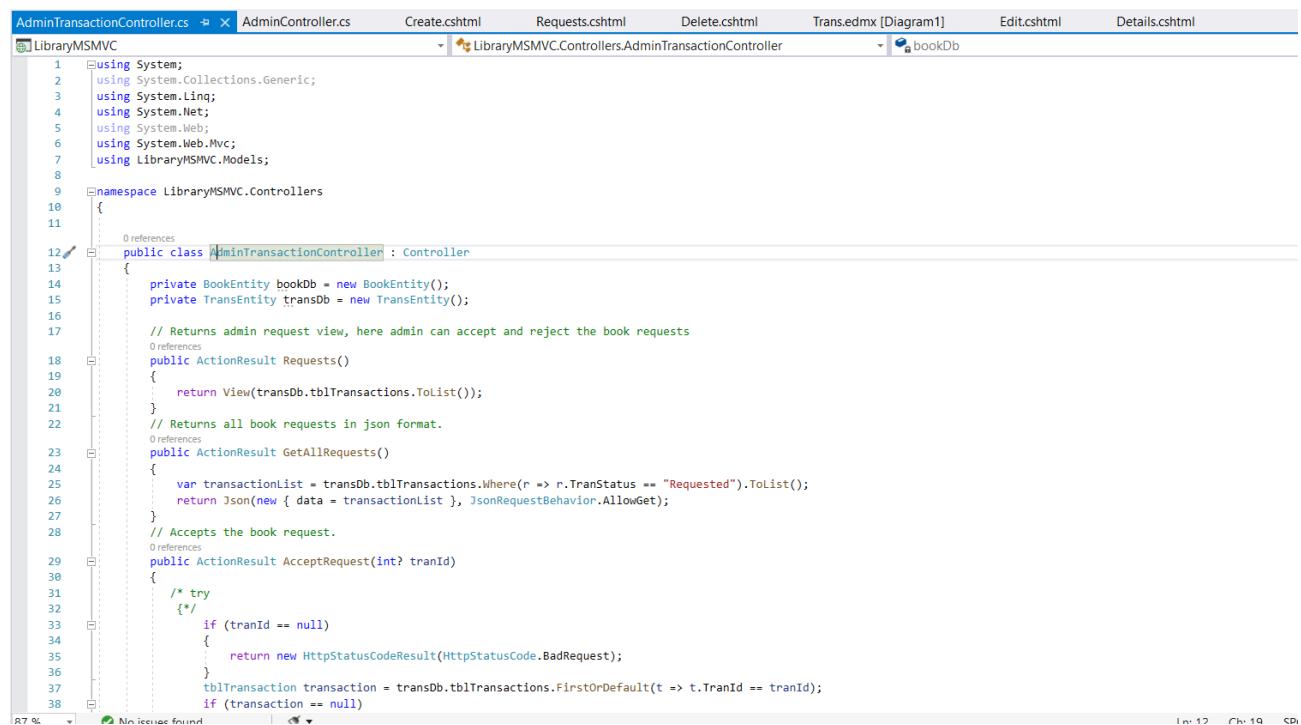


```

AdminController.cs + X Create.cshtml     Return.cshtml     Accepted.cshtml     Requests.cshtml     Delete.cshtml     Trans.edmx [Diagram1]     Edit.cshtml     Details.cshtml
LibraryMSMVC
15  [HttpGet]
16  [HandleError]
17  public ActionResult Login()
18  {
19      return View();
20  }
21
22  // Checks admin credentials, redirecting to admin section (index, tblBooks).
23  [HttpPost]
24  [HandleError]
25  public ActionResult Login(tblAdmin admin)
26  {
27      var adm = adminDb.tblAdmins.SingleOrDefault(a => a.AdminEmail == admin.AdminEmail && a.AdminPass == admin.AdminPass);
28      if (adm != null)
29      {
30          int id = adm.AdminId;
31          Session["adminid"] = adm.AdminId;
32          return RedirectToAction("Index", "TblBooks", new { id = id });
33      }
34      else if (admin.AdminEmail == null && admin.AdminPass == null)
35      {
36          return View();
37      }
38      ViewBag.Message = "User name and password are not matching";
39      return View();
40  }
41  /* Admin credentials validation two
42  public ActionResult Validate(tblAdmin admin)
43  {
44      var adm = db.tblAdmins.SingleOrDefault(a => a.AdminEmail == admin.AdminEmail && a.AdminPass == admin.AdminPass);
45      if (adm != null)
46      {
47          int id = adm.AdminId;
48          return RedirectToAction("Index", "tblBooks", new { @id = id });
49      }
50      else if (admin.AdminEmail == null && admin.AdminPass == null)
51      {
52          return View("Login");
53      }
54      ViewBag.Message = "User name and password are not matching";
}

```

### Admin transaction controller:



```

AdminController.cs + X AdminController.cs     Create.cshtml     Requests.cshtml     Delete.cshtml     Trans.edmx [Diagram1]     Edit.cshtml     Details.cshtml
LibraryMSMVC
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Net;
5  using System.Web;
6  using System.Web.Mvc;
7  using LibraryMSMVC.Models;
8
9  namespace LibraryMSMVC.Controllers
10 {
11
12     public class AdminTransactionController : Controller
13     {
14         private BookEntity bookDB = new BookEntity();
15         private TransEntity transDB = new TransEntity();
16
17         // Returns admin request view, here admin can accept and reject the book requests
18         public ActionResult Requests()
19         {
20             return View(transDB.tblTransactions.ToList());
21         }
22         // Returns all book requests in json format.
23         public ActionResult GetAllRequests()
24         {
25             var transactionList = transDB.tblTransactions.Where(r => r.TranStatus == "Requested").ToList();
26             return Json(new { data = transactionList }, JsonRequestBehavior.AllowGet);
27         }
28         // Accepts the book request.
29         public ActionResult AcceptRequest(int? tranId)
30         {
31             /* try
32             {*/
33             if (tranId == null)
34             {
35                 return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
36             }
37             tblTransaction transaction = transDB.tblTransactions.FirstOrDefault(t => t.TranId == tranId);
38             if (transaction == null)
39             {
40                 return HttpNotFound();
41             }
42             transaction.TranStatus = "Accepted";
43             transDB.SaveChanges();
44             return RedirectToAction("Requests");
45         }
46     }
47 }

```

## Borrow controller:

```

BorrowController.cs ✘ X AdminTransactionController.cs AdminController.cs Create.cshtml Requests.cshtml Delete.cshtml Trans.edmx [Diagram1] Edit.cshtml
LibraryMSMVC
1 using System;
2 using System.Collections.Generic;
3 using System.Data;
4 using System.Data.Entity;
5 using System.Linq;
6 using System.Net;
7 using System.Web;
8 using System.Web.Mvc;
9 using LibraryMSMVC.Models;
10
11 namespace LibraryMSMVC.Controllers
12 {
13     public class BorrowController : Controller
14     {
15         static int userId; // Used to store user id.
16         static string userName; // Used to store user name.
17
18         private UserEntity userDb = new UserEntity();
19         private BookEntity bookDb = new BookEntity();
20         private TransEntity transDb = new TransEntity();
21
22
23         // Returns user books borrow view, here user can request for a book.
24         public ActionResult Index(int? userId, string userName)
25         {
26             if (userId == null)
27             {
28                 return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
29             }
30             tblUser user = userDb.tblUsers.Find(userId);
31             if (user == null)
32             {
33                 return HttpNotFound();
34             }
35
36             BorrowController.userId = (int)userId;
37             BorrowController.userName = userName;
38             return View(bookDb.tblBooks.ToList());
39         }
40     }

```

## Main controller:

```

MainController.cs ✘ X BorrowController.cs AdminTransactionController.cs AdminController.cs Create.cshtml Requests.cshtml Delete.cshtml
LibraryMSMVC
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.Mvc;
6
7 namespace LibraryMSMVC.Controllers
8 {
8     public class MainController : Controller
9     {
10         // Returns home view.
11         public ActionResult Home()
12         {
13             return View();
14         }
15
16         // Returns about view.
17         public ActionResult About()
18         {
19             return View();
20         }
21
22         // Returns contact view.
23         public ActionResult Contact()
24         {
25             return View();
26         }
27
28         // Returns login view.
29         public ActionResult Login()
30         {
31             return View();
32         }
33
34     }
35

```

## User controller:

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.Mvc;
6  using LibraryMSMVC.Models;
7
8  namespace LibraryMSMVC.Controllers
9  {
10     public class UserController : Controller
11     {
12         UserRepository userDb = new UserRepository();
13
14         // Returns user login view, here admin can login.
15         [HttpGet]
16         public ActionResult Login()
17         {
18             return View();
19         }
20
21         // Checks user credentials, redirecting to admin section (index, tblBooks).
22         [HttpPost]
23         public ActionResult Login(tblUser user)
24         {
25             var adm = userDb.tblUsers.SingleOrDefault(a => a.UserEmail == user.UserEmail && a.UserPass == user.UserPass);
26             if (adm != null)
27             {
28                 Session["userId"] = adm.UserId;
29                 Session["username"] = adm.UserName;
30                 return RedirectToAction("Index", "Borrow", new { userId = adm.UserId, userName = adm.UserName });
31             }
32             else if (user.UserEmail == null && user.UserPass == null)
33             {
34                 return View();
35             }
36             ViewBag.Message = "User name and password are not matching";
37             return View();
38         }
39     }

```

87% No issues found | Ln: 1 Ch: 1 SPC

## Table user controller:

```

1  using System;
2  using System.Collections.Generic;
3  using System.Data;
4  using System.Data.Entity;
5  using System.Linq;
6  using System.Net;
7  using System.Web;
8  using System.Web.Mvc;
9  using LibraryMSMVC.Models;
10
11 namespace LibraryMSMVC.Controllers
12 {
13     public class TblUsersController : Controller
14     {
15         private UserRepository userDb = new UserRepository();
16
17         // GET: tblUsers
18         public ActionResult Index()
19         {
20             return View(userDb.tblUsers.ToList());
21         }
22
23         // GET: tblUsers Json
24         public ActionResult GetAll()
25         {
26             var userlist = userDb.tblUsers.ToList();
27             return Json(new { data = userlist }, JsonRequestBehavior.AllowGet);
28         }
29
30         // GET: tblUsers/Details/
31         public ActionResult Details(int? id)
32         {
33             if (id == null)
34             {
35                 return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
36             }
37             tblUser tblUser = userDb.tblUsers.Find(id);
38             if (tblUser == null)

```

87% No issues found | Ln: 1 Ch: 1 SPC

## Table book controller:

```

1  using System;
2  using System.Collections.Generic;
3  using System.Data;
4  using System.Data.Entity;
5  using System.Linq;
6  using System.Net;
7  using System.Web;
8  using System.Web.Mvc;
9  using LibraryMSMVC.Models;
10 
11 namespace LibraryMSMVC.Controllers
12 {
13     public class TblBooksController : Controller
14     {
15         private BookEntity bookDb = new BookEntity();
16 
17         // GET: tblBooks
18         public ActionResult Index()
19         {
20             return View(bookDb.tblBooks.ToList());
21         }
22 
23         // GET: tblBooks/Json
24         public ActionResult GetAll()
25         {
26             var booklist = bookDb.tblBooks.ToList();
27             return Json(new { data = booklist }, JsonRequestBehavior.AllowGet);
28         }
29 
30         // GET: tblBooks/Details/5
31         public ActionResult Details(int? id)
32         {
33             if (id == null)
34             {
35                 return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
36             }
37             tblBook tblBook = bookDb.tblBooks.Find(id);
38             if (tblBook == null)
39             {
40                 return HttpNotFound();
41             }
42             return View(tblBook);
43         }
44     }
45 }

```

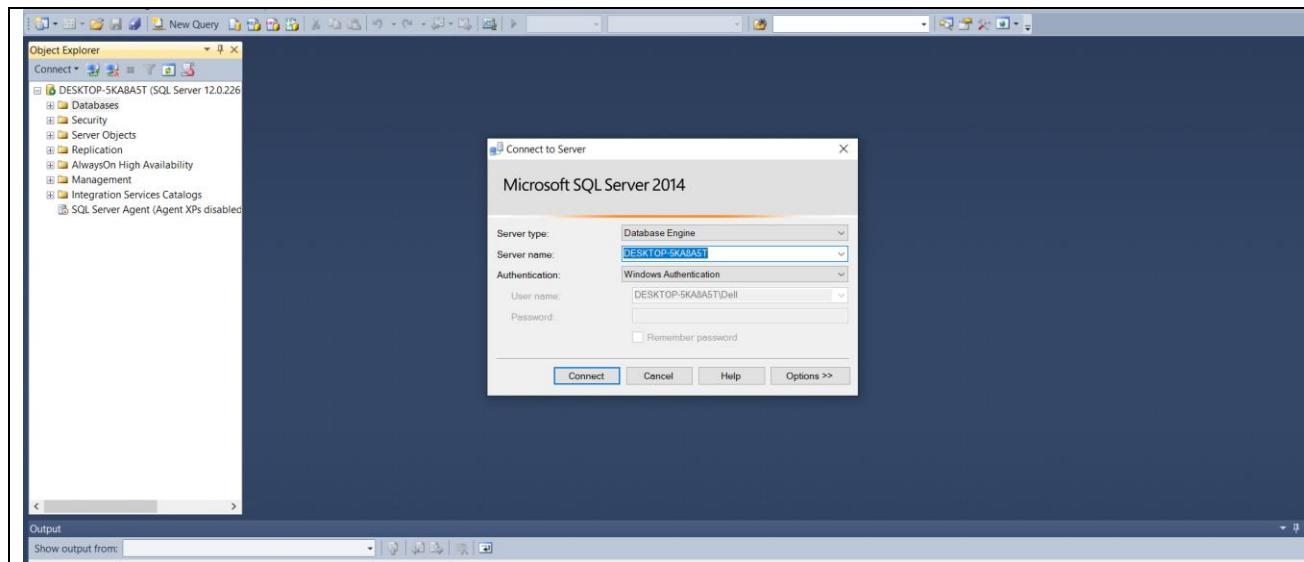
## Web.config

```

<configuration>
    <dependentAssembly>
        <assemblyIdentity name="System.Web.Mvc" publicKeyToken="31bf3856ad364e35" />
        <bindingRedirect oldVersion="1.0.0.0-5.2.7.0" newVersion="5.2.7.0" />
    </dependentAssembly>
    <dependentAssembly>
        <assemblyIdentity name="WebGrease" publicKeyToken="31bf3856ad364e35" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-1.5.2.14234" newVersion="1.5.2.14234" />
    </dependentAssembly>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
        <dependentAssembly>
            <assemblyIdentity name="System.Web.Optimization" publicKeyToken="31bf3856ad364e35" />
            <bindingRedirect oldVersion="1.0.0.0-1.5.2.14234" newVersion="1.5.2.14234" />
        </dependentAssembly>
    </assemblyBinding>
    <runtime>
        <system.codedom>
            <compilers>
                <compiler language="c#;cs;csharp" extension=".cs" type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider, Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
                <compiler language="vb;vbs;visualbasic;vbscript" extension=".vb" type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider, Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
            </compilers>
        </system.codedom>
    </runtime>
    <connectionStrings>
        <add name="AdminEntity" connectionString="metadata=res://*/Models.Admin.csdl|res://*/Models.Admin.ssdl|res://*/Models.Admin.msl;provider=System.Data.SqlClient;provider connection string='Data Source=DESKTOP-1D9C9F0\SQL2019;Initial Catalog=LibraryMSMVC;Integrated Security=True'" />
        <add name="BookEntity" connectionString="metadata=res://*/Models.Book.csdl|res://*/Models.Book.ssdl|res://*/Models.Book.msl;provider=System.Data.SqlClient;provider connection string='Data Source=DESKTOP-1D9C9F0\SQL2019;Initial Catalog=LibraryMSMVC;Integrated Security=True'" />
        <add name="UserEntity" connectionString="metadata=res://*/Models.User.csdl|res://*/Models.User.ssdl|res://*/Models.User.msl;provider=System.Data.SqlClient;provider connection string='Data Source=DESKTOP-1D9C9F0\SQL2019;Initial Catalog=LibraryMSMVC;Integrated Security=True'" />
        <add name="TransEntity" connectionString="metadata=res://*/Models.Trans.csdl|res://*/Models.Trans.ssdl|res://*/Models.Trans.msl;provider=System.Data.SqlClient;provider connection string='Data Source=DESKTOP-1D9C9F0\SQL2019;Initial Catalog=LibraryMSMVC;Integrated Security=True'" />
    </connectionStrings>
    <entityFramework>
        <defaultConnectionFactory type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory, EntityFramework">
            <parameters>
                <parameter value="mssqllocaldb" />
            </parameters>
        </defaultConnectionFactory>
        <providers>
            <provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />
        </providers>
    </entityFramework>

```

## Database screenshots



## Database Tables

The screenshot shows the Object Explorer pane on the left with a tree view of database objects. Under 'dbo.Member', the 'Columns' node is expanded, showing columns: MemberID, MemberName, MemberCategoryID, Adress, ContactNo, Email, Username, Password, and Status. To the right, a detailed view of the 'Member' table is shown in a grid format. The columns are: Column Name (MemberID, MemberName, MemberCategoryID, Adress, ContactNo, Email, Username, Password, Status), Data Type (int, varchar(50), int, varchar(50), varchar(20), varchar(50), varchar(50), varchar(50), varchar(50)), and Allow Nulls (checkboxes all unchecked).

## Create/ Insert Query

The screenshot shows the Object Explorer on the left with the database structure for 'Library Management System'. On the right, the 'SQL Query Editor' window displays the T-SQL code for creating a stored procedure:

```

USE [Library Management System]
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[MEMBER_INSERT]
    @NAME AS VARCHAR(50),
    @CATEID AS INT,
    @ADD AS VARCHAR(50),
    @CONT AS VARCHAR(20),
    @EMAIL AS VARCHAR(50),
    @UNAME AS VARCHAR(50),
    @PNAME AS VARCHAR(50),
    @STATUS AS VARCHAR(50)
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO Member VALUES (@NAME,@CATEID,@ADD,@CONT,@EMAIL,@UNAME,@PNAME,@STATUS)
END

```

The status bar at the bottom indicates "Commands completed successfully."

## Select Query

Two separate SQL query windows are shown:

- SQLQuery3.sql - DE...QF9943\DELL (54)\***: Contains a stored procedure [dbo].[Book\_Select\_By\_BID] which selects a book by BookID.
- SQLQuery4.sql - DE...QF9943\DELL**: Contains a stored procedure [dbo].[Book\_Select] which selects all books from the Book table.

Both windows show the message "Commands completed successfully." in the status bar.

## Delete Query

The screenshot shows the SQL Query Editor window displaying the T-SQL code for creating a stored procedure [dbo].[Book\_Delete] which deletes a book by BookID:

```

USE [Library Management System]
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[Book_Delete]
    @BID AS INT
AS
BEGIN
    SET NOCOUNT ON;
    DELETE FROM Book WHERE BookID=@BID
END

```

The status bar at the bottom indicates "Commands completed successfully."

## Testing

web apps require a person to run them independently from all other applications and work separately as full-function programs. For accurate web application testing, adequate and a set of functionalities are necessary. This testing is complicated as most of the tests are created for a certain setting and hence there are no interactions with other elements. It should be done frequently since it is so important.

### User Interface testing (front-end)

User interface testing is a product/software testing approach that uses the Graphical Users Interface [GUI] to discover faulty conditions. It offers us a realistic perspective of the actual performance and predicted output. It is essential to know defects in user interface codes during the entire software development process, which may be done easily by running the program or interface codes. All application quality depends on the user interface thus it must be properly and efficiently tested.

### Database testing (back-end)

Database testing consists largely of building SQL queries in which the application for which a database is used assets and validates different database actions, structures and properties. In this step the schema may be validated, CRUD operations and transactions tested to ensure correctly set up of the database. These tests can be entirely automated, manual or hybrid with a combination of human and automated procedures. For example, you may go into the database management system and do queries to confirm assumptions in a totally manual test. Or the application UI may be tested to verify if the proper data is returned following an operation.

## Testing methodologies

There are various methods and methodologies which helps to test whole application in proper way depending upon application type.

- **White box or unit testing:** White Box Testing is a test software method where internal structure, software design and code are evaluated to check input flows and to improve design, usability and safety. Software testing. Code is also termed clear box testing, open box testing, transparent box testing, codebased testing and crystal box testing in white box testing. In a white box testing code is visibility for testers.
- **Black box or functional testing:** Black Box Testing is a software test technique where software functionality may be tested without knowledge of the inner structure of the code, specifics of implementation, and internal pathways. The focus of Black Box Testing is largely on software application input and output.

## Testing of system functionalities and performance

The library management preliminarily deals with managing the library of different students. The system contains various functionalities related to library, book, admin, session, category and students. The testing of the system has been carried to determine whether all the above presented designed formswork properly or not. A testing along with actual output is presented below to test the functionalities and performance of different functionalities present in the system.

### Overall system testing of Library management system:

**Admin login testing:** We will try to login into system using username and password. The following picture will show the user information. If we insert user details expect of information on user table then it should be login into LMS system.



The screenshot shows a SQL query window with the following details:

- Query: `Select * from tblAdmins;`
- Results pane showing data from the 'tblAdmins' table:

AdminId	AdminName	AdminEmail	AdminPass
1000	Hari	Hari@ismt.edu.np	Hari@123
1001	saurav	saurav@ismt.edu	saurav@11
2000	shyam	shyam@ismt.edu.np	shyam@123

<b>Tested Data:</b> Admin login	<b>Tested By:</b> Saurav	<b>Date:</b> 2021/12/08
<b>Expected Output:</b> should be login	<b>Result:</b> login Successfully	<b>Remarks:</b> work Successfully

**Justification:** The username [Saurav@gamil.com](mailto:Saurav@gamil.com) is available on our user table of LMS so login is being completed. Hence, our system is working fine.

## Outputs:

Title	Category	Author	Copies	Publication	Pub Name	ISBN	Copyright	Date Added	Status	Actions
Muna Madan	Romance	Kali Baskota	18	Asmita	Asmita Publication	0-892-90282-1	2018	2021-01-20	New	<input type="checkbox"/> Edit <input style="background-color: yellow; color: black; border: none; padding: 0 5px;" type="button" value="Details"/> <input style="background-color: red; color: white; border: none; padding: 0 5px;" type="button" value="Delete"/>
Subtle Art	Action	Jhoeen	10	Syplex	Hunted	0-123-12345-1	2012	11/21/2018	Old	<input type="checkbox"/> Edit <input style="background-color: yellow; color: black; border: none; padding: 0 5px;" type="button" value="Details"/> <input style="background-color: red; color: white; border: none; padding: 0 5px;" type="button" value="Delete"/>
Watch Me	Fiction	Roberto	0	Dunlep	Asmita	0-123-13245-1	2017	11/08/2018	Old	<input type="checkbox"/> Edit <input style="background-color: yellow; color: black; border: none; padding: 0 5px;" type="button" value="Details"/> <input style="background-color: red; color: white; border: none; padding: 0 5px;" type="button" value="Delete"/>

**User login testing:** We will try to login into system using fake username and password. The following picture will show the user information. If we insert user details expect of information on user table then it should not be login into LMS system.

The screenshot shows a SQL Server Management Studio window with the following details:

- Top tabs: DESKTOP-5KA8A5T...A - dbo.tblAdmins, DESKTOP-5KA8A5T...SWA - dbo.Table\_1, SQLQuery1.sql - DE...5KA8A5T\...Dell (52)\*
- Query window: A single line of SQL code: "Select \* from tblUsers;"
- Results pane: Displays a table with 7 rows of data. The columns are: UserId, UserName, UserGen..., UserD..., UserAdm..., UserEmail, and UserPass.
- Data in Results pane:

UserId	UserName	UserGen...	UserD...	UserAdm...	UserEmail	UserPass
1	bikeshbhaila	Male	CS	2015350	bikesh@gmail.com	bikesh@123
2	Saurav	Male	Mech	2016309	saurav@gmail.com	saurav@123
3	Nabin	Male	Civil	20170612	nabin@gmail.com	nabin@123
4	abik	Male	Ec	2017912	abik.mushyakho94@gmail.com	P@ssw0rd
5	rekha11	Male	Ec	5	rekha11@gmail.com	nabin123@
7						

<b>Tested Data:</b> User login	<b>Tested By:</b> Saurav	<b>Date:</b> 2021/12/08
<b>Expected Output:</b> should not login	<b>Result:</b> denied login	<b>Remarks:</b> work perfectly

**Justification:** The username [Sandip@gmail.com](mailto:sandip@gmail.com) is not available on our user table of LMS so login was not completed. Hence, our system is working fine.

The screenshot shows a login form with the following fields and message:

- Form title: Login
- Fields:
  - UserEmail: sandip@gmail
  - UserPass: (redacted)
- Message: User name and password are not matching
- Submit button: Login

**Insertion Test:** Here we insert the book on the table to see whether it works successfully or not.

<b>Tested Data:</b> Insertion Test	<b>Tested By:</b> Saurav	<b>Date:</b> 2021/12/08
<b>Expected Output:</b> should Insert book and its details	<b>Result:</b> Book is inserted successfully	<b>Remarks:</b> Book added successfully

**Justification:** Here we add the details of book on LMS so it was completed. Hence, system is working fine.

**Inputs:**

The screenshot shows the 'Add book' form in the Orland LMS application. The form fields are as follows:

- Title: Computer
- Category: Science
- Author: Franklin
- Copies: 16
- Publication: Syplex & Saya
- Pub Name: Joe Rogan
- ISBN: 0-123-12345-12
- Copyright: 1980
- Date Added: 12/09/2021
- Status: Old

A 'Submit' button is located at the bottom of the form. Below the form, there is a link labeled 'Back'.

**Outputs:**

The screenshot shows the 'Books' index page in the Orland LMS application. A modal window displays the message "localhost:44312 says Book added successfully". The main table lists the following books:

Category	Author	Copies	Publication	Pub Name	ISBN	Copyright	Date Added	Status	Actions
BookCategory	BookAuthor	BookCopies	BookPub	BookPubName	BookISBN	Copyright	DateAdded	Status	
3000	Hindi	Fiction	Kali Shrestha	9	Shyanya	Shyanya Publication	0-892-90282-1	2019	12/24/2021 Old
3001	Subtle Art	Action	Jhooen	10	Syplex	Hunted	0-123-12345-1	2012	11/21/2018 Old
3002	Watch Me	Fiction	Roberto	0	Dunle	Asmita	0-123-13245-1	2017	11/08/2018 Old
3003	Cmac	Drama	Rogean	10	Syplex	Syplexpublication	0-123-12345-1	1900	12/01/2021 Old
3004	Cmac	Drama	Rogean	10	Syplex	Publication	0-123-12345-1	1900	12/01/2021 Old
3005	Computer	Science	Franklin	16	Syplex	Joe Rogan	0-123-12345-1	NULL	12/09/2021 Old

**Database Output:** After we fill on the UI/UX, we also need to check out on the database to make assure that testing is going perfectly.

```
40 %
Results Messages
| Select * from tblBooks;

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BookId | BookTitle | BookCategory | BookAuthor | BookCopies | BookPub | BookPubName | BookISBN | Copyri... | DateAdded | Status |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 3000 | Hindi | Fiction | Kali Shrestha | 9 | Shyanya | Shyanya Publication | 0-892-90282-1 | 2019 | 12/24/2021 | Old |
| 3001 | Subtle Art | Action | Jhooen | 10 | Syplex | Hunted | 0-123-12345-1 | 2012 | 11/21/2018 | Old |
| 3002 | Watch Me | Fiction | Roberto | 0 | Dunle | Asmita | 0-123-13245-1 | 2017 | 11/08/2018 | Old |
| 3003 | Cmac | Drama | Rogean | 10 | Syplex | Syplexpublication | 0-123-12345-1 | 1900 | 12/01/2021 | Old |
| 3004 | Cmac | Drama | Rogean | 10 | Syplex | Publication | 0-123-12345-1 | 1900 | 12/01/2021 | Old |
| 3005 | Computer | Science | Franklin | 16 | Syplex | Joe Rogan | 0-123-12345-1 | NULL | 12/09/2021 | Old |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

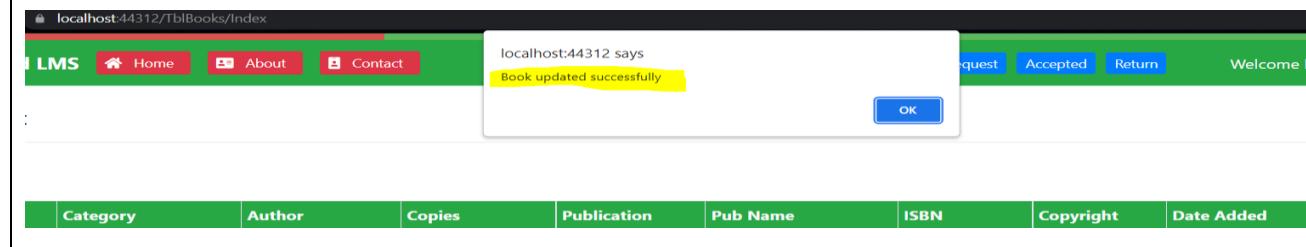
**Update test:** Here we have update the book table.

<b>Tested Data:</b> Insertion Test	<b>Tested By:</b> Saurav	<b>Date:</b> 2021/12/08
<b>Expected Output:</b> should Insert book and its details	<b>Result:</b> Book is inserted successfully	<b>Remarks:</b> Book added successfully

**Justification:** Here we add the details of book on LMS so it was completed. Hence, system is working fine.

### Inputs:

### Outputs:



**Database Output:** After we updates the inputs of books, we will also see whether It will works or not in Database

```
update tblBooks set BookTitle='Nepali', BookCopies=12 where Bookid=3000;
select * from tblBooks;
```

BookId	BookTitle	BookCateg...	BookAuthor	BookCopi...	BookPub	BookPubName	BookISBN	Copyri...	DateAdded	Status	
1	3000	Nepali	Fiction	Kali Shrestha	12	Shyakya	Shyakya Publication	0-892-90282-1	2019	12/24/2021	Old
2	3001	Subtle Art	Action	Jhoen	10	Syplex	Hunted	0-123-12345-1	2012	11/21/2018	Old
3	3002	Watch Me	Fiction	Roberto	0	Dunlep	Asmita	0-123-13245-1	2017	11/08/2018	Old
4	3003	Cmac	Drama	Rogean	10	Syplex	Syplexpublication	0-123-12345-1	1900	12/01/2021	Old
5	3004	Cmac	Drama	Rogean	10	Syplex	Publication	0-123-12345-1	1900	12/01/2021	Old
6	3005	Computer	Science	Franklin	16	Syplex	Joe Rogan	0-123-12345-1	NULL	12/09/2021	Old

**Deletion Test:** Here, we will try to delete the book on LMS

<b>Tested Data:</b> Deletion Test	<b>Tested By:</b> Saurav	<b>Date:</b> 2021/12/08
<b>Expected Output:</b> should deleted books	<b>Result:</b> Book is deleted successfully	<b>Remarks:</b> Book deleted successfully

**Justification:** Here we add the details of book on LMS so it was completed. Hence, system is working fine.

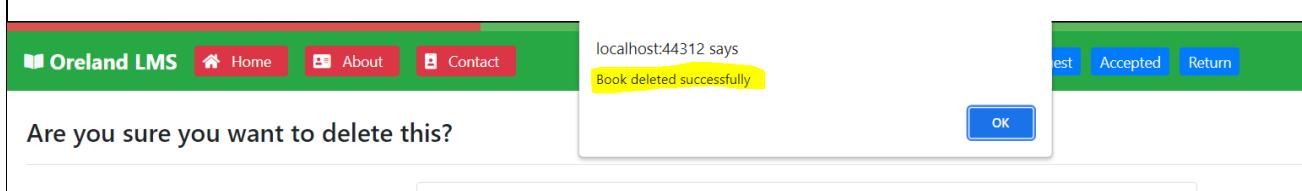
### Inputs:

The screenshot shows a confirmation dialog box asking "Are you sure you want to delete this?". Below the dialog, a book's details are displayed in a grid:

Title Computer	Category Science
Author Franklin	Copies 16
Publication Syplex	Pub Name Joe Rogan
ISBN 0-123-12345-1	Copyright
Date Added 12/09/2021	Status Old

A red "Delete" button is visible at the bottom left of the dialog.

### Outputs:



**Database Output:** As, we don't see the book title belongs to computer in the database. So, it is confirmed that testing is done successfully.

```
SELECT * FROM tblBooks;
```

The screenshot shows the results of the above query in SSMS. The table has the following columns: BookId, BookTitle, BookCateg..., BookAuthor, BookCopi..., BookPub, BookPubName, BookISBN, Copyri..., DateAdded, Status. The data is as follows:

	BookId	BookTitle	BookCateg...	BookAuthor	BookCopi...	BookPub	BookPubName	BookISBN	Copyri...	DateAdded	Status
1	3000	Nepali	Fiction	Kali Shrestha	12	Shyakya	Shyakya Publication	0-892-90282-1	2019	12/24/2021	Old
2	3001	Subtle Art	Action	Jhoeen	10	Syplex	Hunted	0-123-12345-1	2012	11/21/2018	Old
3	3002	Watch Me	Fiction	Roberto	0	Dunlep	Asmita	0-123-13245-1	2017	11/08/2018	Old
4	3004	Cmac	Drama	Rogean	10	Syplex	Publication	0-123-12345-1	1900	12/01/2021	Old

**Add user testing:** Here, we test the request send by user in LMS.

<b>Tested Data:</b> User add Test	<b>Tested By:</b> Saurav	<b>Date:</b> 2021/12/08
<b>Expected Output:</b> should add user	<b>Result:</b> User is added Successfully	<b>Remarks:</b> Added successfully

**Justification:** Here we add the User on LMS so it was completed. Hence, system is working fine.

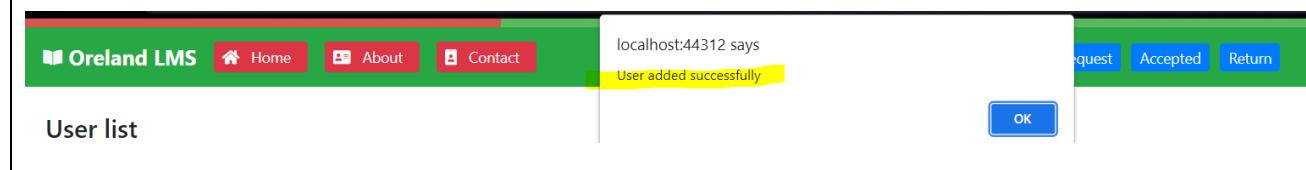
#### Inputs:

Add user

The screenshot shows a form titled "Add user". It includes fields for UserName (containing "Hari"), UserGender (with radio buttons for Male, Female, and Other, where Male is selected), UserDep (containing "Mech"), UserAdmNo (containing "10"), UserEmail (containing "hari@gmail.com"), and UserPass (containing "hari@123"). A "Submit" button is located at the bottom of the form.

Back

#### Outputs:



**Database output:** We have added user in LMS, to assure the test we also retrieve the data in database which depict that user has been added in database

```
Select * from tblUsers;
```

	UserId	UserName	UserGen...	UserD...	UserAdm...	UserEmail	UserPass
1	bikeshbhaila	Male	CS	2015350		bikesh@gmail.com	bikesh@123
2	Saurav	Male	Mech	2016309		saurav@gmail.com	saurav@123
3	Nabin	Male	Civil	20170612		nabin@gmail.com	nabin@123
4	abik	Male	Ec	2017912		abik.mushyakho94@gmail.com	P@ssw0rd
5	rekha11	Male	Ec	5		rekha11@gmail.com	nabin123@
6	Hari	Male	Mech	10		hari@gmail.com	hari@123

**Form validation Testing:** we have done form validating testing on book to see whether it includes

<b>Tested Data:</b> Form validation Test	<b>Tested By:</b> Saurav	<b>Date:</b> 2021/12/08
<b>Expected Output:</b> Shouldn't load the ambiguous data	<b>Result:</b> depict the data isn't being loaded	<b>Remarks:</b> successfully done

**Justification:** Here we add attributes on book items on LMS. We see ambigous data isnnot being loaded due to validation on form.

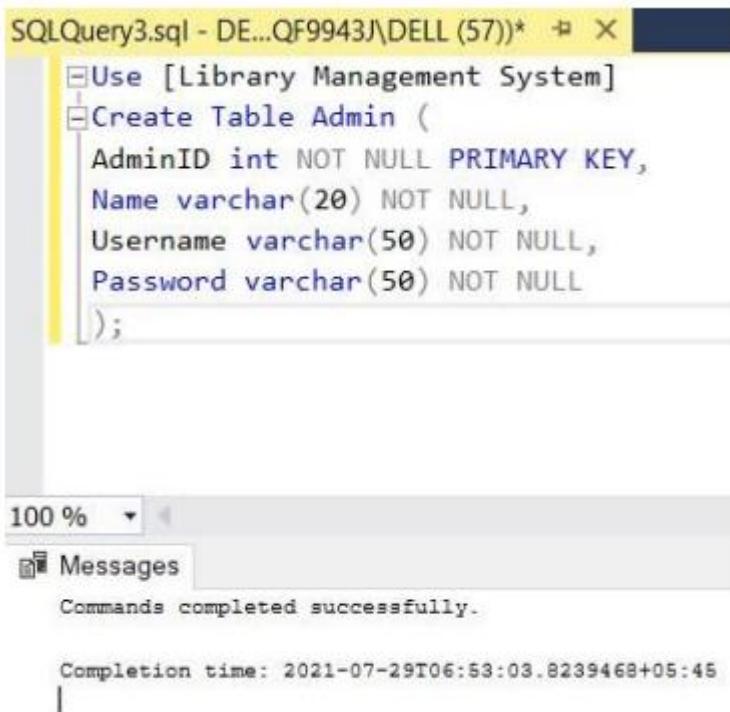
<p>Title  <input type="text" value="Subtitle of art"/></p> <p>The Title field is invalid</p> <p>Author  <input type="text" value="kali prasad baskota"/></p> <p>The Author field is invalid</p> <p>Publication  <input type="text" value="Shyakya"/></p> <p>ISBN  <input type="text" value="0-123-12345-12"/></p> <p>The ISBN field is invalid</p> <p>Date Added  <input type="text" value="11/21/2018"/></p> <p><input checked="" type="checkbox"/> Submit</p>	<p>Category  <input type="text" value="Drama"/></p> <p>Copies  <input type="text" value="15"/></p> <p>The Copies field is required.</p> <p>Pub Name  <input type="text" value="Syplex publication"/></p> <p>The Pub Name field is invalid</p> <p>Copyright  <input type="text" value="1980"/></p> <p>Status  <input type="text" value="New"/></p>
---	---

## Database Testing

### Primary Key Constraint testing

<b>Tested Data:</b> Execute the query for primary key constraint	<b>Tested By:</b> Saurav	<b>Date:</b> 2021/12/08
<b>Expected Output:</b> Primary key should be created	<b>Result:</b> Primary key was created successfully.	<b>Remarks:</b> successfully done

**Justification:** Here we add Primary Key on database table.



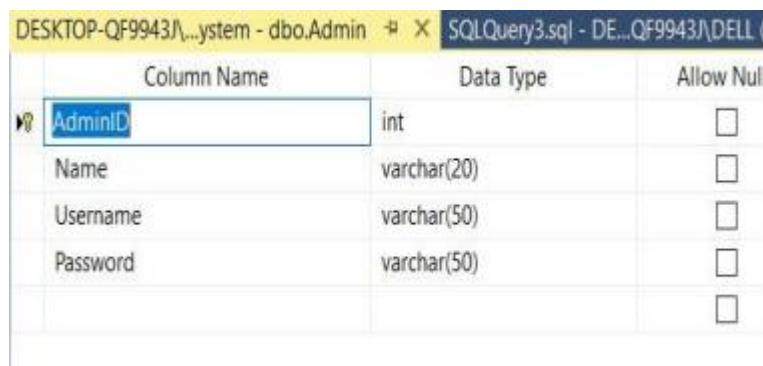
```
SQLQuery3.sql - DE...QF9943J\DELL (57)* ↻ X
Use [Library Management System]
Create Table Admin (
    AdminID int NOT NULL PRIMARY KEY,
    Name varchar(20) NOT NULL,
    Username varchar(50) NOT NULL,
    Password varchar(50) NOT NULL
);
```

100 %

Messages

Commands completed successfully.

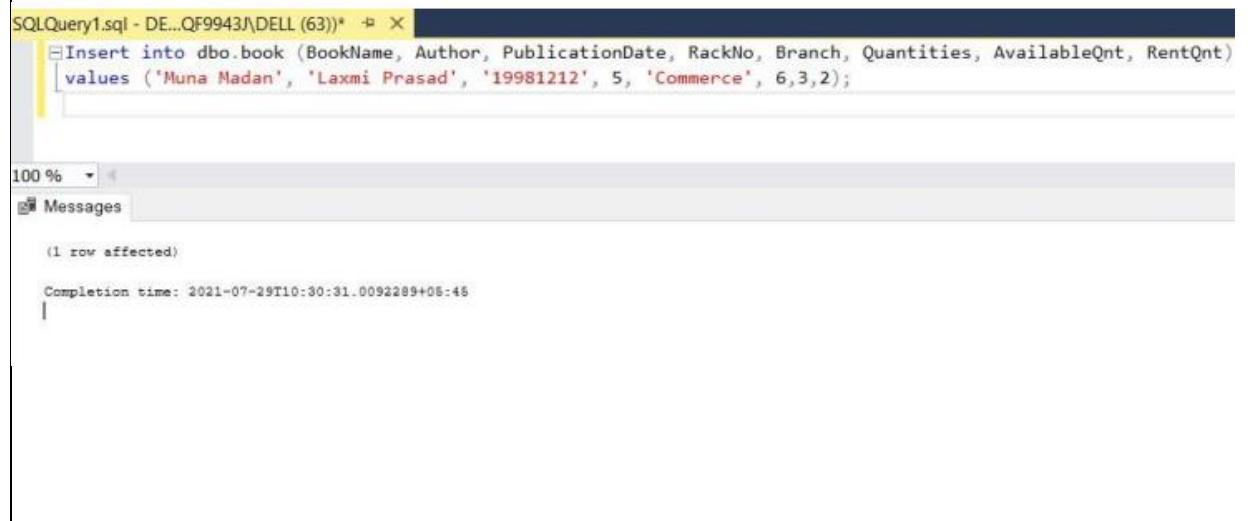
Completion time: 2021-07-29T06:53:03.8239468+05:45



Column Name	Data Type	Allow Null
AdminID	int	<input type="checkbox"/>
Name	varchar(20)	<input type="checkbox"/>
Username	varchar(50)	<input type="checkbox"/>
Password	varchar(50)	<input type="checkbox"/>
		<input type="checkbox"/>

**Inserting testing**

<b>Tested Data:</b> Execute query for data insertion in any table.	<b>Tested By:</b> Saurav	<b>Date:</b> 2021/12/08
<b>Expected Output:</b> Data should be inserted into the allocated table.	<b>Result:</b> Data was added successfully.	<b>Remarks:</b> successfully done
<b>Justification:</b> Here we Execute query for data insertion in any table.		



The screenshot shows a SQL query being run in SQL Server Management Studio. The query is:

```
Insert into dbo.book (BookName, Author, PublicationDate, RackNo, Branch, Quantities, AvailableQnt, RentQnt)
values ('Muna Madan', 'Laxmi Prasad', '19981212', 5, 'Commerce', 6,3,2);
```

The results show 1 row affected and a completion time of 2021-07-29T10:30:31.0092289+05:45.

**Select Testing**

<b>Tested Data:</b> Select all or any data from any table and execute the query.	<b>Tested By:</b> saurav	<b>Date:</b> 2021/12/08
<b>Expected Output:</b> Selected data will appear in the table.	<b>Result:</b> Selected data was seen in the table	<b>Remarks:</b> work perfectly

**Justification:** As, we have Select data from table and execute the query.

The screenshot shows a SQL Server Management Studio interface. At the top, there are tabs for 'Results' and 'Messages'. Below the tabs is a table with the following data:

	BookID	BookName	Author	PublicationDate	RackNo	Branch	Quantities	AvailableQnt	RentQnt
1	1	Muna Madan	Laxmi Prasad	1998-12-12	5	Commerce	6	3	2

Below this table, the status bar shows the connection details: DESKTOP-5KA8A5T...A - dbo.tblAdmins and DESKTOP-5KA8A5T...SWA - dbo.Table\_1. To the right of the status bar, it says SQLQuery1.sql - DE...5KA8A5T\DEll (52)\*.

In the main query pane, the following SQL code is written:

```
Select * from tblBooks;
```

Below the query pane, another table is displayed with the following data:

	BookId	BookTitle	BookCateg...	BookAuthor	BookCopi...	BookPub	BookPubName	BookISBN	Copyri...	DateAdded	Status
1	3000	Nepali	Fiction	Kali Shrestha	12	Shyakya	Shyakya Publication	0-892-90282-1	2019	12/24/2021	Old
2	3001	Subtle Art	Action	Jhoeen	10	Syplex	Hunted	0-123-12345-1	2012	11/21/2018	Old
3	3002	Watch Me	Fiction	Roberto	0	Dunlep	Asmita	0-123-13245-1	2017	11/08/2018	Old
4	3004	Cmac	Drama	Rogean	10	Syplex	Publication	0-123-12345-1	1900	12/01/2021	Old

## Update Testing

<b>Tested Data:</b> Enter update query to make any changes in the table.	<b>Tested By:</b> Saurav	<b>Date:</b> 2021/12/08
<b>Expected Output:</b> Data of the allocated table should be updated.	<b>Result:</b> Required data was updated.	<b>Remarks:</b> work perfectly
<b>Justification:</b> Here, we enter update query to make any changes in the table.		

```
update tblBooks set BookTitle='Nepali', BookCopies='12' where Bookid=3000;
Select * from tblBooks;
```

140 % < Results Messages

	BookId	BookTitle	BookCateg...	BookAuthor	BookCopi...	BookPub	BookPubName	BookISBN	Copyri...	DateAdded	Status
1	3000	Nepali	Fiction	Kali Shrestha	12	Shyakya	Shyakya Publication	0-892-90282-1	2019	12/24/2021	Old
2	3001	Subtle Art	Action	Jhoeen	10	Syplex	Hunted	0-123-12345-1	2012	11/21/2018	Old
3	3002	Watch Me	Fiction	Roberto	0	Dunlep	Asmita	0-123-13245-1	2017	11/08/2018	Old
4	3004	Cmac	Drama	Rogean	10	Syplex	Publication	0-123-12345-1	1900	12/01/2021	Old

### Deletion Testing

<b>Tested Data:</b> Enter delete query to delete any data from the table	<b>Tested By:</b> Saurav	<b>Date:</b> 2021/12/08
<b>Expected Output:</b> Data of the allocated table should be deleted.	<b>Result:</b> Assigned data were deleted.	<b>Remarks:</b> work perfectly
<b>Justification:</b> Here, we delete query to delete any data from table.		

```
Delete from tblBooks where BookId=3004;
Select * from tblBooks;
```

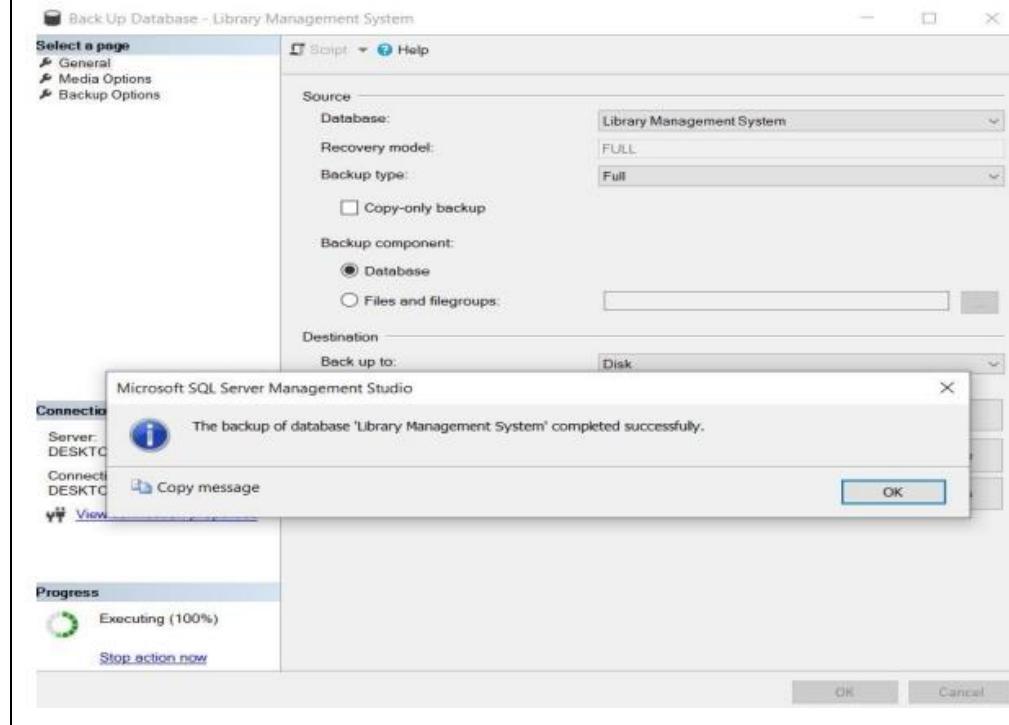
140 % < Results Messages

	BookId	BookTitle	BookCateg...	BookAuthor	BookCopi...	BookPub	BookPubName	BookISBN	Copyri...	DateAdded	Status
1	3000	Nepali	Fiction	Kali Shrestha	12	Shyakya	Shyakya Publication	0-892-90282-1	2019	12/24/2021	Old
2	3001	Subtle Art	Action	Jhoeen	10	Syplex	Hunted	0-123-12345-1	2012	11/21/2018	Old
3	3002	Watch Me	Fiction	Roberto	0	Dunlep	Asmita	0-123-13245-1	2017	11/08/2018	Old

## Database Backup Testing

<b>Tested Data:</b> Right click on database and on task option click on database backup option.	<b>Tested By:</b> saurav	<b>Date:</b> 2021/12/08
<b>Expected Output:</b> Selected database should be backed up.	<b>Result:</b> Database was backed up successfully.	<b>Remarks:</b> work perfectly

**Justification:** here, we do right click on database and on task option click on database backup option.



## Database Restoration Testing

<b>Tested Data:</b> Right click on databases and choose restore database option and select the database that you want to restore.	<b>Tested By:</b> saurav	<b>Date:</b> 2021/12/08
<b>Expected Output:</b> Selected database should be restored successfully.	<b>Result:</b> Library Management System database was restored successfully.	<b>Remarks:</b> work perfectly
<b>Justification:</b> Here, we click right on databases and choose restore database option and select the database that you want to restore.		

## **Implementing effective features in LMS to handle concurrency, security, authorizations and data recovery**

The LMS application that we have developed may not have all the effective feature required for the complete user and client requirement. The features like concurrency, security, authorizations and data recovery should be implemented for the betterment of LMS application. Database maintenance is a set of tasks that are performed with an objective to improve the database so designed. It helps to check the faults and errors in database, checking the smoothness of database and backing up files for future use. There are routines meant to help performance, free up disk space, check for data errors, check for hardware faults, update internal statistics, and many other obscure (but important) things. Database security refers to the measures applied to protect and secure database or stored procedures from unauthorized access, intentional or accidental threats. A database can be protected and secured using various security mechanisms.

One important aspect of maintaining a database is simply backing up the data so that, if anything happens, there will be another copy available. Some database systems actually do this automatically, sending a backup to another location every day, every week, or within any other set period of time. Backups are usually not enough, however. Database maintenance includes checking for signs of corruption in the database, looking for problem areas, rebuilding indexes, removing duplicate records, and checking for any abnormalities in the database that might signal a problem. The goal is to keep the database operating smoothly for users, so that ideally they never need to think about maintenance issues. A database that is not maintained can become sluggish, and people may start to experience problems when trying to access records. Below I have provided some of the database routines which were used for Library management system:-

**Data Concurrency Control:** The standard definition of a transaction states that “every query that runs in a SQL Server is in a transaction,” that means any query we run on a SQL Server is considered as being in a transaction. It could either be a simple SELECT query or any UPDATE or ALTER query. If we run a query without mentioning the BEGIN TRAN keyword then it would be considered an implicit transaction. If we run a query that starts with BEGIN TRAN and ends with COMMIT or ROLLBACK, then it would be considered an explicit transaction.

### **Database Maintenance**

Database maintenance aids in smoothing running of the system using various routines. Some of the categories of database routines that are used are provided below:

➤ **Transactions:** A transaction is a set of changes that must be made together and are being performed by a single unit, which reads or updates the contents of the database. It is a logical unit of work on the database. This may be an entire program, a piece of a program or a single command (like the SQL commands such as INSERT or UPDATE) and it may engage in any number of operations on the database. A database transaction must be atomic, consistent, isolated and durable. It is important to control the transactions to ensure the data integrity and to handle database errors. The following commands are used to control transactions.

1. **Atomicity:** A transaction must be fully complete, saved or completely undone (rolled back). A sale in a retail store database illustrates a scenario which explains atomicity, e.g., the sale consists of an inventory reduction and a record of incoming cash. Both either happen together or do not happen - it's all or nothing.
2. **Consistency:** The transaction must be fully compliant with the state of the database as it was prior to the transaction. In other words, the transaction cannot break the database's constraints. For example, in our system database table any phone number column can only contain numerals, then consistency dictates that any transaction attempting to enter an alphabetical letter may not commit.
3. **Isolation:** Transaction data must not be available to other transactions until the original transaction is committed or rolled back.
4. **Durability:** Transaction data changes must be available, even in the event of database failure.

The command used in transactions are:-

Command	Action
Set transaction	It places a name on transaction
Rollback	To rollback to previous changes.
Commit	To save changes
Save point	To create the checkpoint within transactions group in which to roll back from.

```

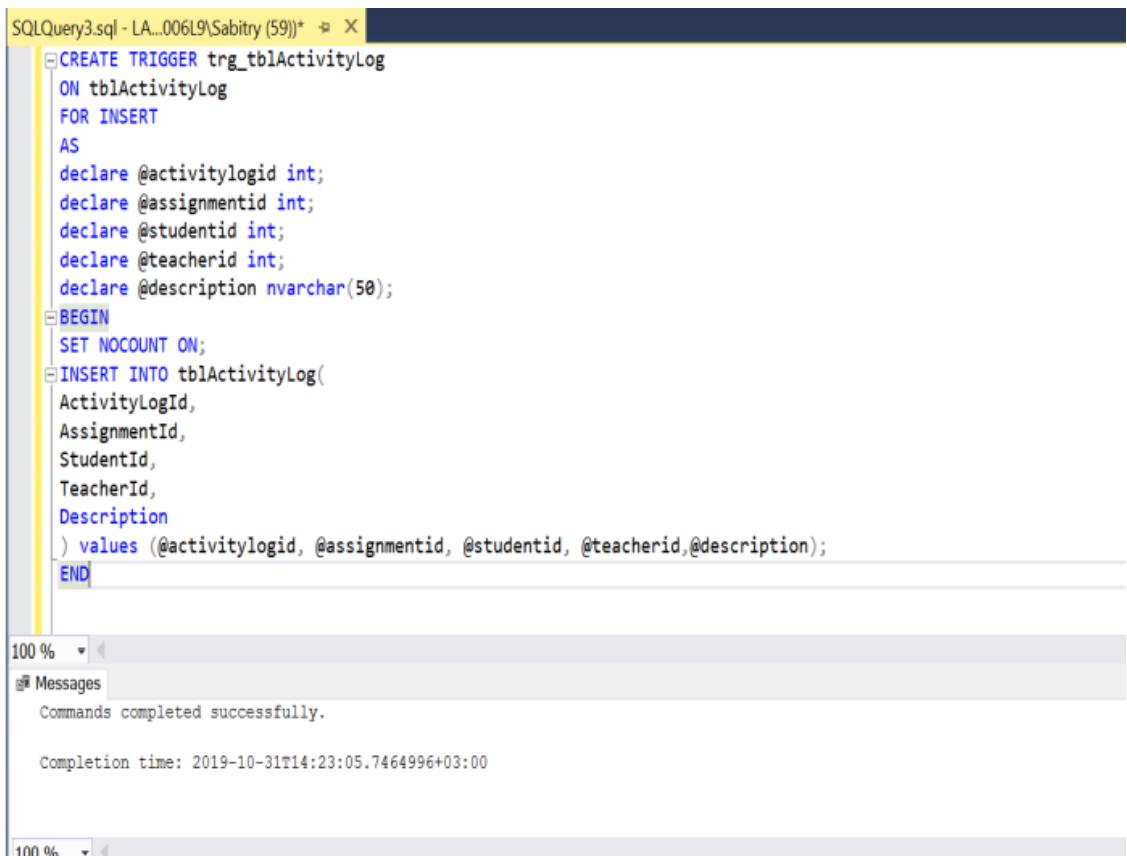
]ELSE IF (@Flag = 'IssueBook')
]    BEGIN
]        If Exists (Select 'x' from Books where BookID = @BookId)
]        begin
]            Select '1' as ErrorCode, 'Book Doesnot exists' as Msg, null id
]        end
]
]        SET @noofBooks =(Select Count(NoOfBooks) from IssueBook where UserId =@UserId)
]
]        If @noofBooks >5
]        begin
]            Select '1' as ErrorCode, 'User Cannot Have More than 5 Books' as Msg, null id
]        end
]
]        Else
]            set @UserId = (select UserId from Users where UserName =@User)
]            Begin Transaction
]            Insert into IssueBook
]                (BookId,[UserId],BookIssuedDate,IsIssuedVerified
]            )
]            Values
]                (@Book,@UserId,GetDate(),0
]            )
]            Commit transaction
]            select '0' as ErrorCode, 'Requested to issue book' as Msg
]        END

```

*Figure: Transaction*

- **Triggers:** A trigger is a stored procedure in database, which automatically invokes whenever a special event in the database occurs. Triggers can be defined on the table, view, schema, or database with which the event is associated. They help the database designer ensure certain actions, such as maintaining an audit file, are completed regardless of which program or user makes changes to the data. For example, a trigger can be invoked when a row is inserted into a specified table or when certain table columns are being updated. The commands used in trigger are:-

Create trigger[trigger_name]	Creates or replaces an existing trigger with the trigger_name.
[before   after]	This specifies when the trigger will be executed
{insert   update   delete}	This specifies the DML operation.
on [table_name]	This specifies the name of the table associated with the trigger.
[for each row]	This specifies a row-level trigger, i.e., the trigger will be executed for each row being affected.
[trigger_body]	This provides the operation to be performed as trigger is fired



```

CREATE TRIGGER trg_tblActivityLog
ON tblActivityLog
FOR INSERT
AS
declare @activitylogid int;
declare @assignmentid int;
declare @studentid int;
declare @teacherid int;
declare @description nvarchar(50);
BEGIN
SET NOCOUNT ON;
INSERT INTO tblActivityLog(
ActivityLogId,
AssignmentId,
StudentId,
TeacherId,
Description
) values (@activitylogid, @assignmentid, @studentid, @teacherid,@description);
END

```

100 % ▾

Messages

Commands completed successfully.

Completion time: 2019-10-31T14:23:05.7464996+03:00

100 % ▾

*Figure: Triggers*

- **Stored Procedures:** Stored procedures are the prepared SQL code so that code can be reused repeatedly. Instead of having to write that query each time, we would save it as a stored procedure and then just call the stored procedure to execute the SQL code that we saved as part of the stored procedure (Robidoux, 2016). Based on the parameters that are passed, stored procedures act accordingly and saves a lot of time.

```

CREATE OR ALTER PROCEDURE PROC_PUBLISHER
(
    @Flag      NVARCHAR(100)  =NULL,
    @PublicationName NVARCHAR(100) =NULL,
    @RowId      BIGINT        =NULL,
    @User       NVARCHAR(100)  =NULL,
    @Address    NVARCHAR(100)  =NULL,
    @PhoneNo   NVARCHAR(100)  =NULL
)

AS
BEGIN
SET NOCOUNT ON;
BEGIN TRY
IF (@Flag = 'Insert')
BEGIN
    INSERT INTO Publishers(PublicationName,Address,PhoneNumber,createdBy,CreatedDateLocal,CreatedDateUTC) VALUES(@PublicationName,@Address,@PhoneNo,@User,GETDATE(),GETUTCDATE())
    SELECT '0' AS ErrorCode, 'Successfully Inserted Publication' AS Message, NULL ID
END

ELSE IF (@Flag = 'Update')
BEGIN
    IF NOT EXISTS (SELECT * FROM Publishers WHERE RowId = @RowId)
    BEGIN
        SELECT '1' AS ErrorCode, 'Publisher Id Not Found' AS Message, NULL ID
    END
    UPDATE Publishers SET PublicationName = @PublicationName, Address = @Address, PhoneNumber = @PhoneNo, UpdatedBy = @User, UpdatedDateLocal = GETDATE(), UpdatedDateUTC = GETUTCDATE()
    WHERE RowId = @RowId
    SELECT '0' AS ErrorCode, 'Successfully Updated Publication' AS Message, NULL ID
END

ELSE IF (@Flag = 'List')
BEGIN
    SELECT RowId, PublicationName,Address,PhoneNumber FROM Publishers WITH (NOLOCK)
END

ELSE IF (@Flag = 'GetById')
BEGIN
    SELECT RowId,PublicationName,Address,PhoneNumber FROM Publishers WITH (NOLOCK) WHERE RowId = @RowId
END

ELSE IF (@Flag = 'Delete')
BEGIN
    DELETE FROM Publishers WHERE RowId = @RowId
END
END TRY
BEGIN CATCH
    IF (@@ERROR > 0)
    BEGIN
        ROLLBACK TRANSACTION;
        SELECT '1' AS ErrorCode,
        ERROR_MESSAGE() AS Msg,
        ERROR_LINE() ID;
    END;
END CATCH;
End

```

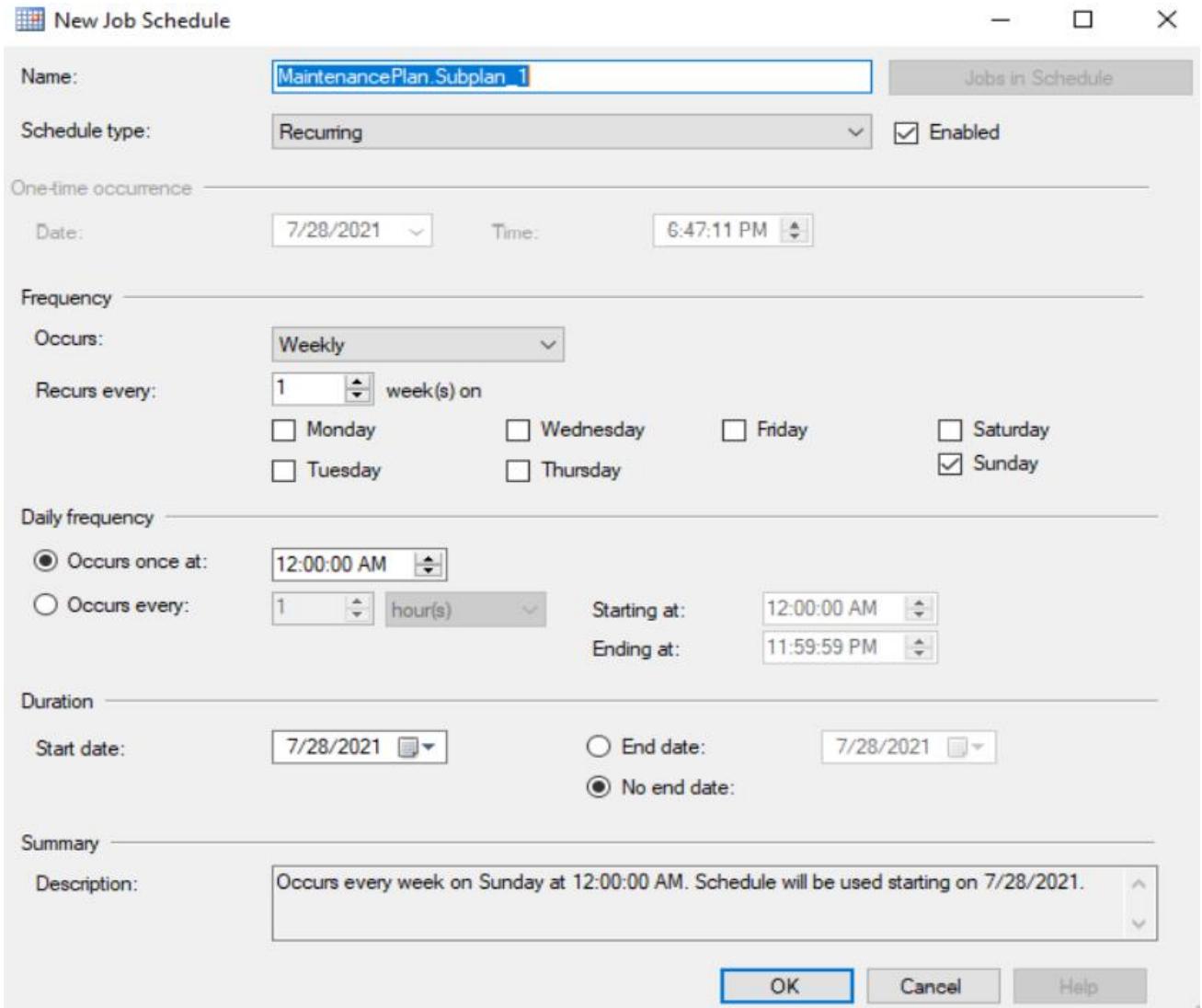
Figure: Store procedure

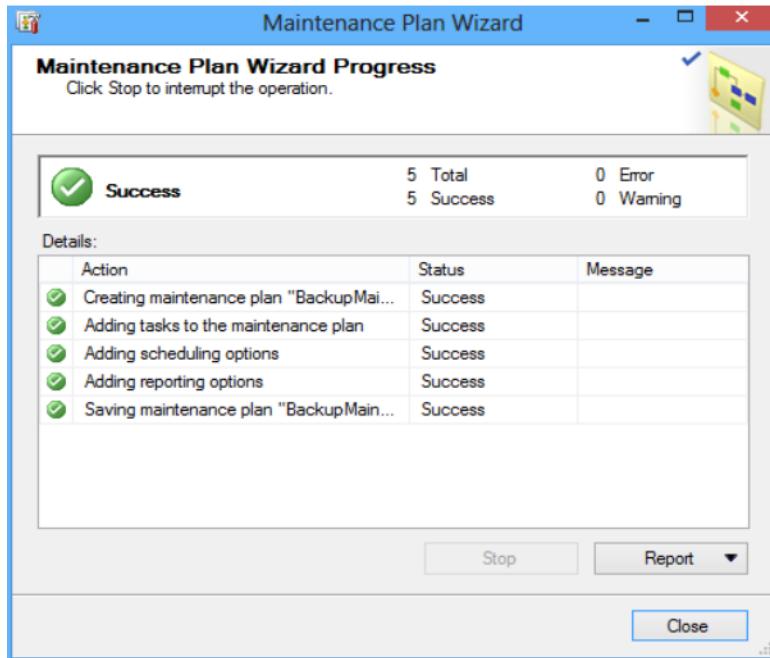
## Database Security

Database security are the mechanisms that are performed in order to secure and protect the data. The various security measures that are applied in MS SQL server for protecting the data are described below:

**Data Backup and Recovery:** Data backup and recovery is the beneficial and the safest route and tool for protecting the organization past, present and future as it allows to store the data separately and recover them from catastrophic damage and failure (TechTarget, 2016). In SQL server, various types of backups can be performed in order to keep the data safe.

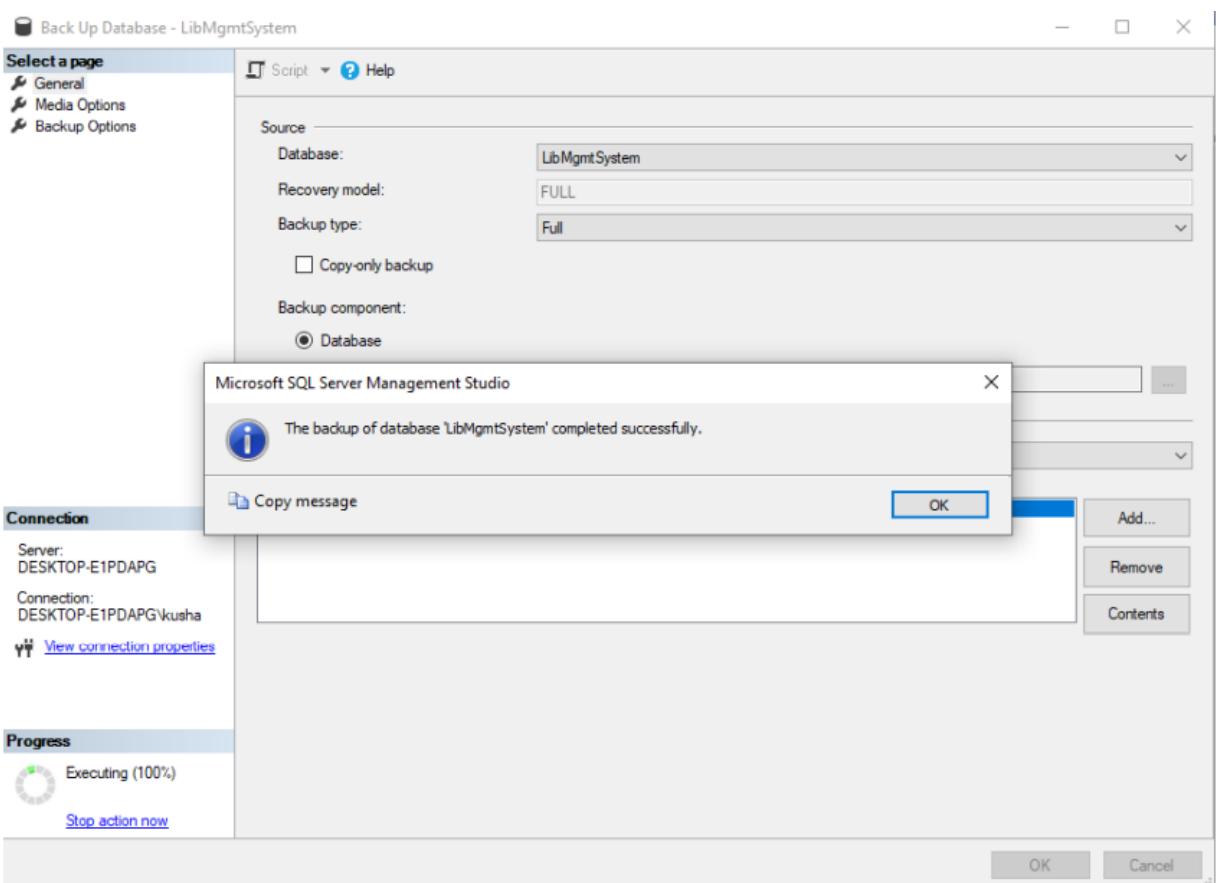
- **Scheduled backups:** This helps to back up the data automatically according to the schedule time period as provided by the users. This helps to keep the data safe on timely basis.

A screenshot of the "New Job Schedule" dialog box. The window title is "New Job Schedule". The "Name:" field contains "MaintenancePlan.Subplan\_1". The "Schedule type:" dropdown is set to "Recurring", and the "Enabled" checkbox is checked. Under "One-time occurrence", the date is set to "7/28/2021" and the time is "6:47:11 PM". The "Frequency" section shows "Weekly" selected, with "1 week(s) on" chosen. Days of the week are listed with checkboxes: Monday, Wednesday, Friday, Saturday, Tuesday, Thursday, and Sunday, where Sunday is checked. The "Daily frequency" section has two options: "Occurs once at: 12:00:00 AM" (radio button selected) and "Occurs every: 1 hour(s)" (radio button not selected). To the right, there are fields for "Starting at: 12:00:00 AM" and "Ending at: 11:59:59 PM". The "Duration" section includes "Start date: 7/28/2021" and "End date: 7/28/2021" (radio button selected), with "No end date:" also available. The "Summary" section contains a "Description:" field with the text "Occurs every week on Sunday at 12:00:00 AM. Schedule will be used starting on 7/28/2021." At the bottom right are "OK", "Cancel", and "Help" buttons.

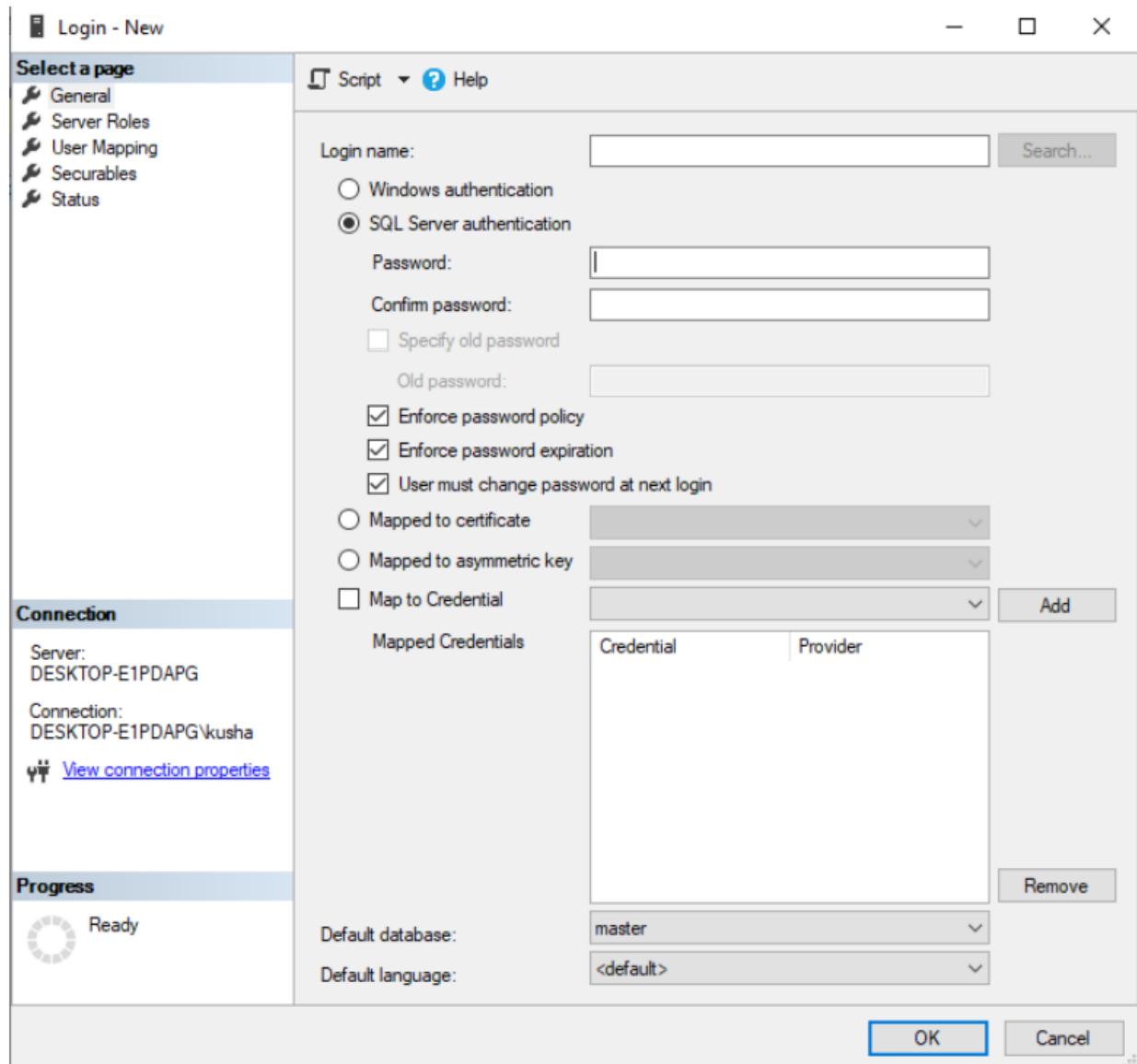


*Figure: Scheduled Backups*

- **Backup database as SQL server dumps:** This helps to back up the data of a single database as a whole. It allows users to back up the data as per the choice. Using this back up, users can only keep the back up of either structure, data or of both data and structure.



- **Authentication and authorization:** Authorization and authentication are the access control mechanisms that are used to allow only authentic users to get access to database (TutorialsPoint, 2016). SQL server has a feature to keep restrictions and prevent particular user from accessing the data. Under the security features, it allows to maintain various permissions that protect the data from being modified from unauthorized users. Login in database is prohibited through username/password combination.



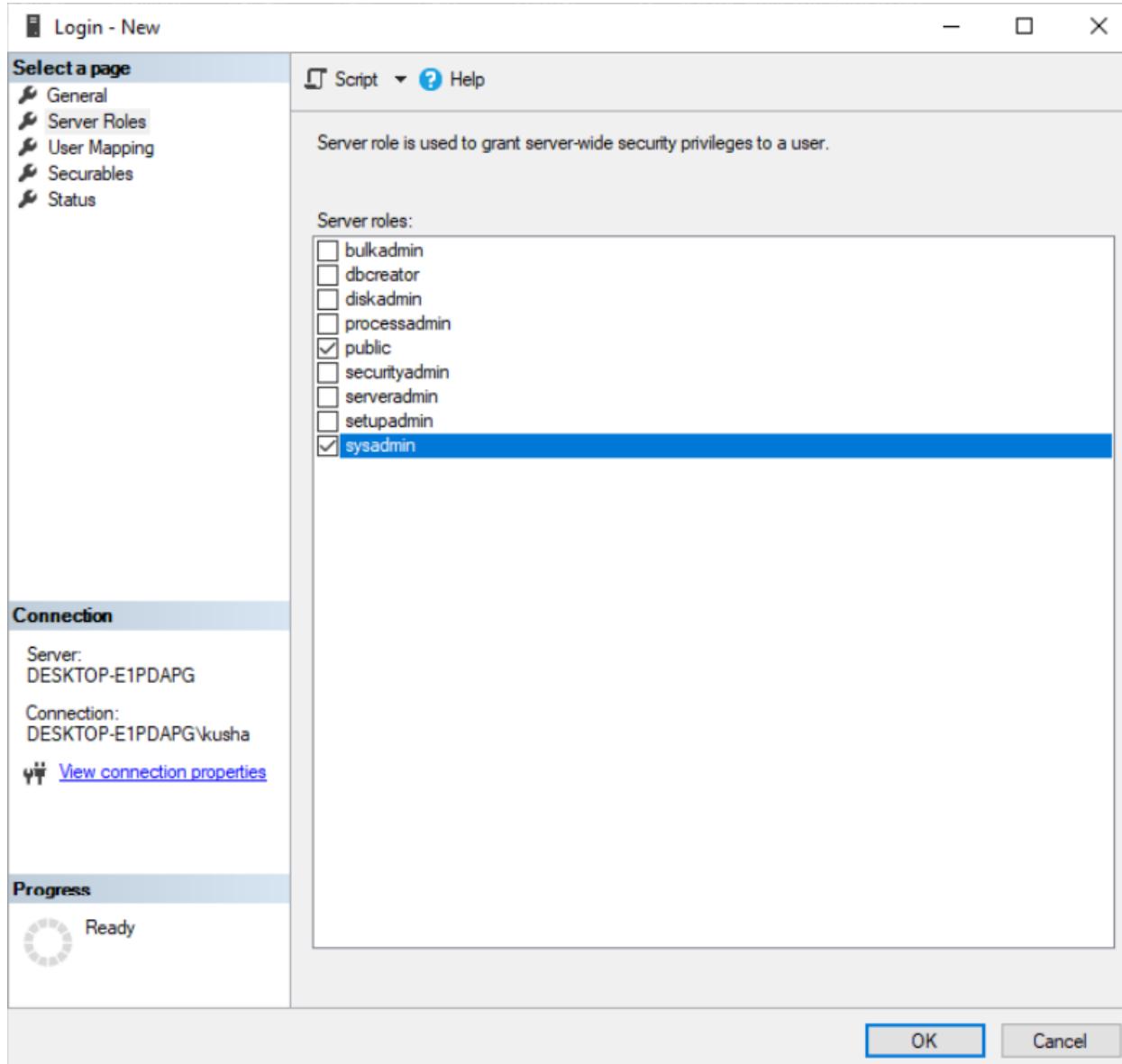


Figure: Authentication and Authorization

### Evaluation of effectiveness of designed system against client requirements

The library management system was designed using C# as programming based on .NET framework using Visual Studio as IDE. Similarly, SQL server was used as the database management tool for storing the data. The effectiveness of the designed system was evaluated comparing with the client requirements. The users and relation database system included following requirements:

- Any library member should be able to search books by their title, author, and subject category as well by the publication date.
- Each book will have a unique identification number and other details including a rack number which will help to physically locate the book.
- There could be more than one copy of a book, and library members should be able to checkout and reserve any copy. We will call each copy of a book, a book item.
- The system should be able to retrieve information like who took a particular book or what are the books checked-out by a specific library member.
- There should be a maximum limit (5) on how many books a member can check-out.
- There should be a maximum limit (10) on how many days a member can keep a book.
- The system should be able to collect fines for books returned after the due date.
- Members should be able to reserve books that are not currently available.
- Each book and member card will have a unique id. The system will be able to read book id from books and members' library cards.
- Email notification should be delivered on the available of the reserved books to those particular students.

The proper evaluation of the system was carried out to verify whether all the modules would work properly or not. All the modules involved in the library management system such as User, Books, Fines, Category, Publisher, Author, IssueBooks and Reservations etc. functioned properly and smoothly. No any errors and exceptions were present in the system that would compromise the requirements of the clients. Comparing the system requirements with the client requirements it was clear that most of the client requirements well addressed well by the designed system. Whenever the necessary details are filled in any forms, the message pops out showing the successful entry of any information. Similarly, entry of unnecessary information shows error. This can be clearly examined from above figures. In database software, the creation of separate table and use of different validations like data type validation, constraints validation prevented data redundancy and anomalies aiding in normalization process. The referential integrity has

been maintained using foreign keys. All the tables are linked with each and duplication is prohibited. Data consistency and integrity are routinely examined using various maintenance tools and security measures are applied well. All such activities provide a proper instance of system effectiveness which fulfilled all the users and system requirements successfully.

Moreover, in the database management software i.e. MS SQL I have properly used and created all the tables with the proper use of validations like data type validation, constraints validation to prevent the data redundancy and anomalies like update, delete and insert anomalies. I have also used the foreign keys in tables where I needed to use that table specific column in another table which also helped to maintain maintain the referential integrity. All the tables are linked with each and duplication is prohibited. Data consistency and integrity are routinely examined using various maintenance tools and security measures are applied well. All such activities provide a proper instance of system effectiveness which fulfilled all the users and system requirements successfully. Furthermore to increase the security in the database also I have implemented the authentication and authorization feature provided by SQL server to make sure that none other than the system administrator is able to modify and have access to the database of library management system. I also have made sure that if somehow the data of whole system gets deleted or destroyed then it is available in the backup file. I have implemented the scheduled backup and the whole backup for the system. The scheduled backup is enabled at five in the evening which is the time to close the college and at that time the database will automatically backup all the data that were added that day in database. In this way after doing the proper testing of system and evaluation analysis I think that all the client requirements are fulfilled in library management system. The system has been developed the way that now the college administration can keep track of all the books borrow by students while also tracking their process i.e. if someone has submitted the books or not or if it is feedback process or it is already checked. So I can say that all the features developed in the system meets the client requirements documentation provided by the college administration.

## Benefits & Effectiveness of System

In the above documentation I have evaluated the developed features of the application with the client requirements and below I have provided the benefits of the library management system for college administration:-

- Any library member should be able to search books by their title, author, and subject category as well by the publication date.
- Each book will have a unique identification number and other details including a rack number which will help to physically locate the book.
- There could be more than one copy of a book, and library members should be able to check-out and reserve any copy. We will call each copy of a book, a book item.
- The system should be able to retrieve information like who took a particular book or what are the books checked-out by a specific library member.
- There should be a maximum limit (5) on how many books a member can check-out.
- There should be a maximum limit (10) on how many days a member can keep a book.
- The system should be able to collect fines for books returned after the due date.
- Members should be able to reserve books that are not currently available.
- Each book and member card will have a unique id. The system will be able to read book id from books and members' library cards.
- Email notification should be delivered on the availability of the reserved books to those particular students.

After pointing out the various benefits which library management system I have also evaluated the performance of the system based on various aspects which are described below:-

**Speed:** While developing the application I have followed the best coding standards and code is clean and functions properly. I have implemented the exceptions handling using try catch block and some of the code is reused properly using various object oriented programming features so the speed of the application is good.

**User Experience:** As it is the web application the design and GUI part of the application is not much that needed. But I have provided up to par level of graphics using various fonts and color for the application. The application is going to be handled by the college Library coordinator and staffs of college so I think the experience of application is good as the other users like customers aren't involved. As it is not the web application and various difficult features aren't involved I don't think

that the application will need the user documentation. Therefore in overall though it is web application because of using fonts and colors in forms graphics is upto par and user experience is also good.

**Security:** The Library management system is developed with the implementation of strong authentication and authorization feature. When the user first try to access the application then s/he have to provide the user login credentials and without the proper credentials they can't have access to the application. There are two user credentials in the login form i.e. admin and staff so when the user provides admin username and password s/he will be directed to admin dashboard which have all the features. If user provides the staff credentials then s/he will be directed to staff dashboard which is authorized with less features. Therefore the Library management system web application has good security measures implemented for the advanced security of application.

## Conclusion

The design of different diagrams as per the requirements of the clients helped to develop system in easier way. All the requirements of the college were addressed successfully. The diagrams helped to provide the overview of the system. With the help of ER diagram, it was easier to identify the different entities present in the system and their relationship were clearly known. Similarly, the class diagram, data flow diagram, schema diagram helps to depict the system work flow that help to address the requirements of the clients. The designs helped to optimize the performance of the system providing the overview of how the system works and what factors are to be addressed well for the proper development of the software. The effectiveness of the system was checked examining all the modules and their functionalities. The various factors that affect the system was examined and appropriate results was obtained that provided evidence that all the modules and their functionalities work well.

**[Part 3]**

Create a lab report: Demonstrate the system administration and management tools available on the chosen platform

- Demonstrate the tools available in the system to monitor and optimize system performance, and examine the audit logs.
- Demonstrate the tools available in the system to manage security and authorizations.
- Assess the effectiveness of the system administration and management tools available on the platform identifying any shortcomings of the tools.
- Assess any future improvements that may be required to ensure the continued effectiveness of the database system.

## Introduction

An administration has wide variety of tools in order to ensure proper system performance. Functions such as user management, system monitoring, backup and recovery, and access control are performed by system administration with the help of variety of management tools available on the selected platform.

In the previous part of my assignment, I have discussed about the various tools available in the database software MS SQL like authentication and authorization, schedule backup etc. to ensure the security and authorization in database. To complete the final part of my assignment I am going to demonstrate the system administration and management tools available in my database software. I am also going to assess the effectiveness of the system administration and those tools while identifying the shortcomings also.

## System Monitoring and Optimizing Tools

The system monitoring tools are used to continuously keep track of the status of the system in use, in order to have the earliest warning of failures, defects or problems and to improve them. These tools also help in detecting unusual and unauthorized, internal or external activities while still gauging the effectiveness of security tools and policies in place. In so doing, system administrators are able to improve the prevention and protection of sensitive data from intruders. These tools also allow relevant authorities to detect, identify and take corrective measures against threats and attacks. Depending on the configuration of those tools, database administrator may be able to reconstruct data or restore it to a previous state. By capturing, keeping logs, analyzing, and alerting on policy violations without interfering with the systems' performance, such system monitoring technologies give us with real-time monitoring and data protection. MS SQL Server has a number of tools that aid in the monitoring of data system, as well as optimize the system's performance in the following ways:-

## Features & Capabilities:

- Database performance monitoring
- Database issue detection
- SQL query analysis
- Change tracking
- Query tuning
- Query reporting

## Database Performance Monitoring Comparison:

When considering a database performance monitoring tool, consider the following aspects of each product offering.

**Query Analysis:** Database performance monitoring comparisons have varying levels of query analysis support. Although nearly all tools have simple query analysis and documentation, automatic query tuning is not always available. Database performance management software with that emphasis could be suitable for companies looking for assistance cleaning up their queries.

**Support Needs:** Such database performance management tools are available as a subscription with continuing support, and others are available as permanent licenses with restricted support. Consider a software as a service option or a permanent license for long-term support if the company requires continuing support.

**Cloud and on premises support:** Any database output management systems are made for on-premises applications, and others are made specifically for cloud databases. If your company exclusively uses one or the other, make sure you have a database monitoring application that works with the one you're using. Hybrid cloud systems are supported by several database performance management software.

## Database Monitoring and optimization tool:

- **Solar Winds Database Performance Analyzer:** SolarWinds Database Performance Analyzer (DPA) is a popular tools among network administrators due to its unique take on database monitoring, which recognizes database response time as the most valuable performance metric.
- **Paessler PRTG Network Monitor:** Oracle, PostgreSQL, Microsoft SQL Server, and MySQL can all be monitored with Paessler PRTG Network Monitor. It entails much more than just keeping track of databases. PRTG can track a variety of metrics, including but not limited to:
  - (1) Cloud-based applications
  - (2) Web-based applications
  - (3) Packets, traffic, IP addresses, and other network metrics
- **Site24x7:** Site24x7 is not a database management service. Instead, it's a cloud-based management service with network, device, and server monitoring capabilities. A database output optimizer is included in the tool's server management kit.
- **SQL Power Tool:** SQL Power Tools describes itself as a database management tool with “zero impact.” A straightforward and lightweight tool provides users with a broad overview of their

database's results. Despite its simplicity, the tool has a lot of influence. It can catch any SQL operation thanks to its SQL text and measurement of end-user reaction time. Every SQL assertion can be benchmarked using SQL Power Tools.

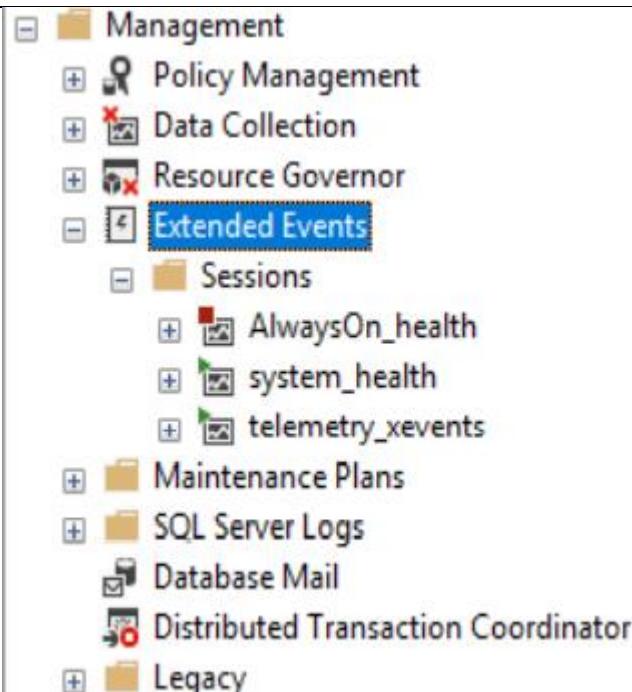
- **Redgate SQL Monitor:** Redgate SQL Monitor has an excellent alert system, with 40 alerts available right out of the box, including memory and CPU use problems, SQL bugs, insufficient disk space, and more. The updates can be customized as well.

## Monitoring and optimization in Library Management System

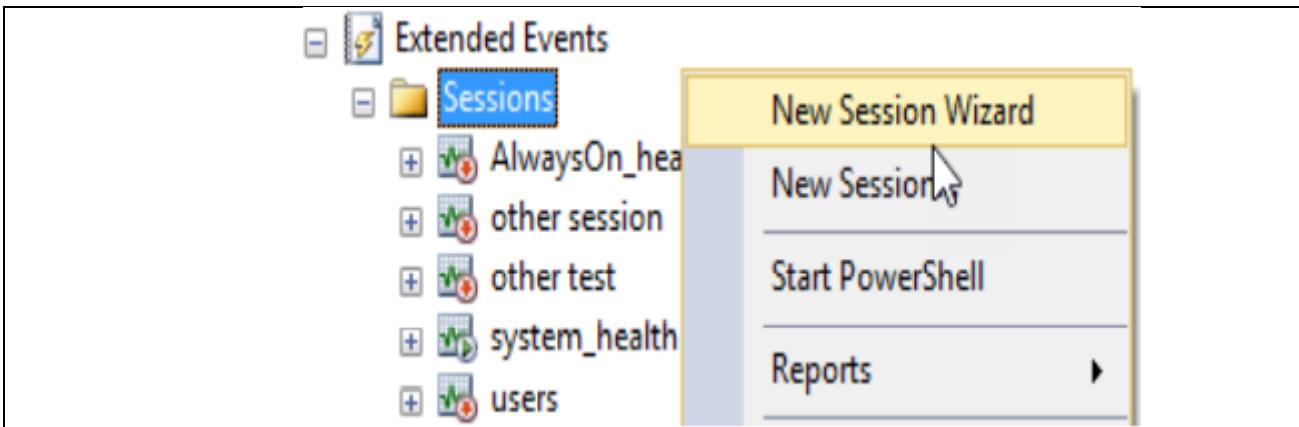
### Extended Events

The extended events architecture in SQL Server allows users to gather as much or as little data as they need to debug or detect a performance issue. Extended Events is very flexible and scalable. It's a simple performance monitoring solution that consumes very little processing power. The new session wizard and new session provide two graphical user interfaces for creating, modifying, displaying, and analyzing your session data. I've included a report on how to use extended events in your library management system below: -

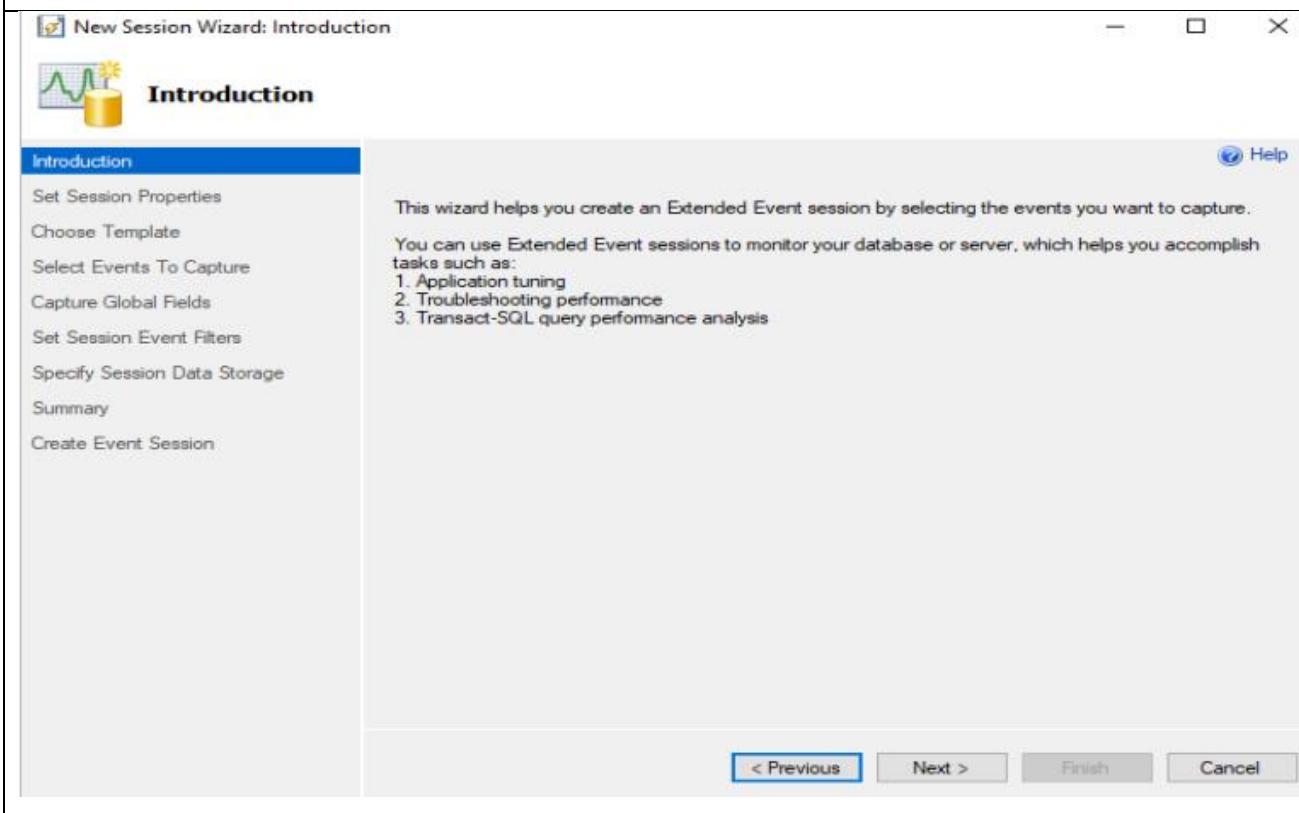
**Step 1:** After opening the sql server go to the object explorer and expand management where we can see extended events option. Then expand the extended events options and we can see another option i.e. Sessions as shown in the figure below.



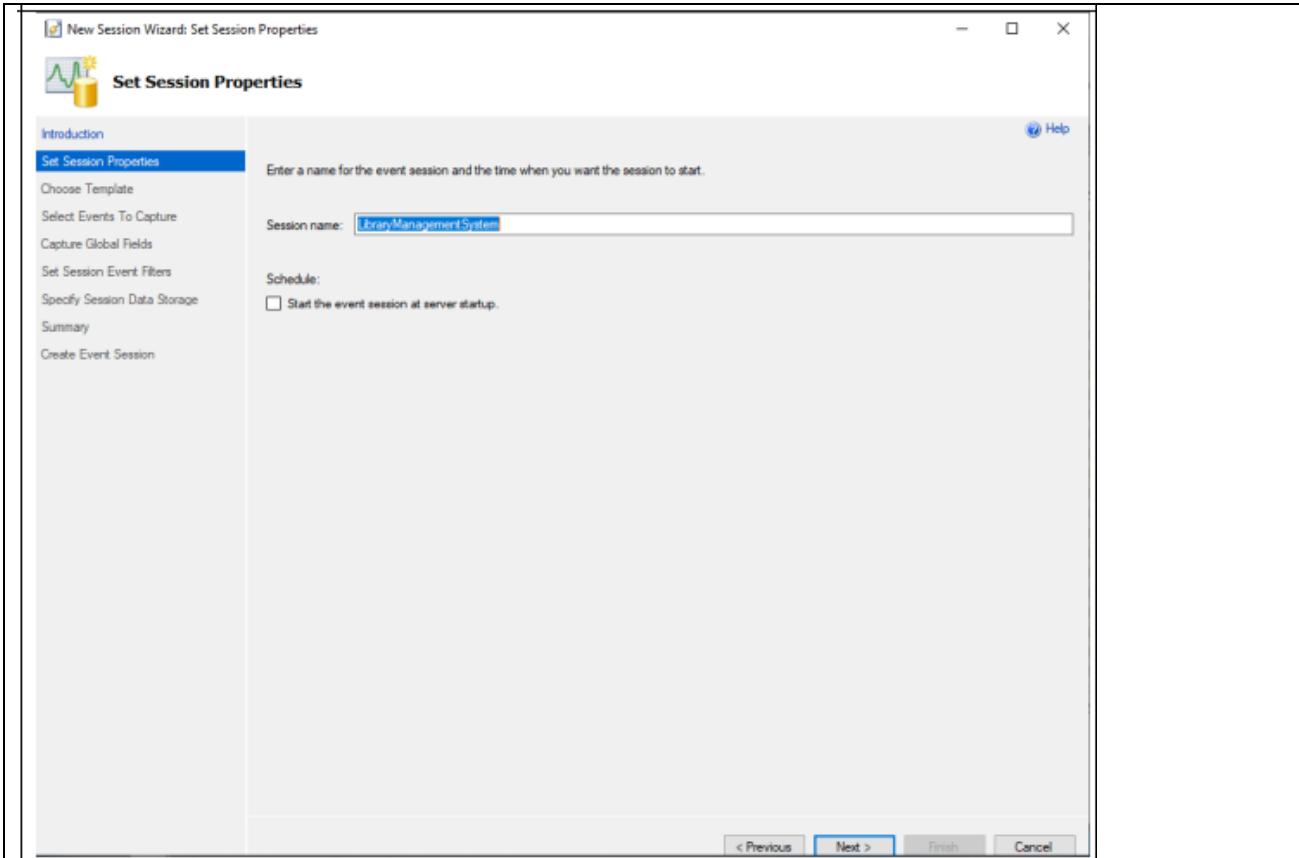
**Step 2:** After that right click on the session option available just below the extended events option. Then select the new session wizard option in option bar.



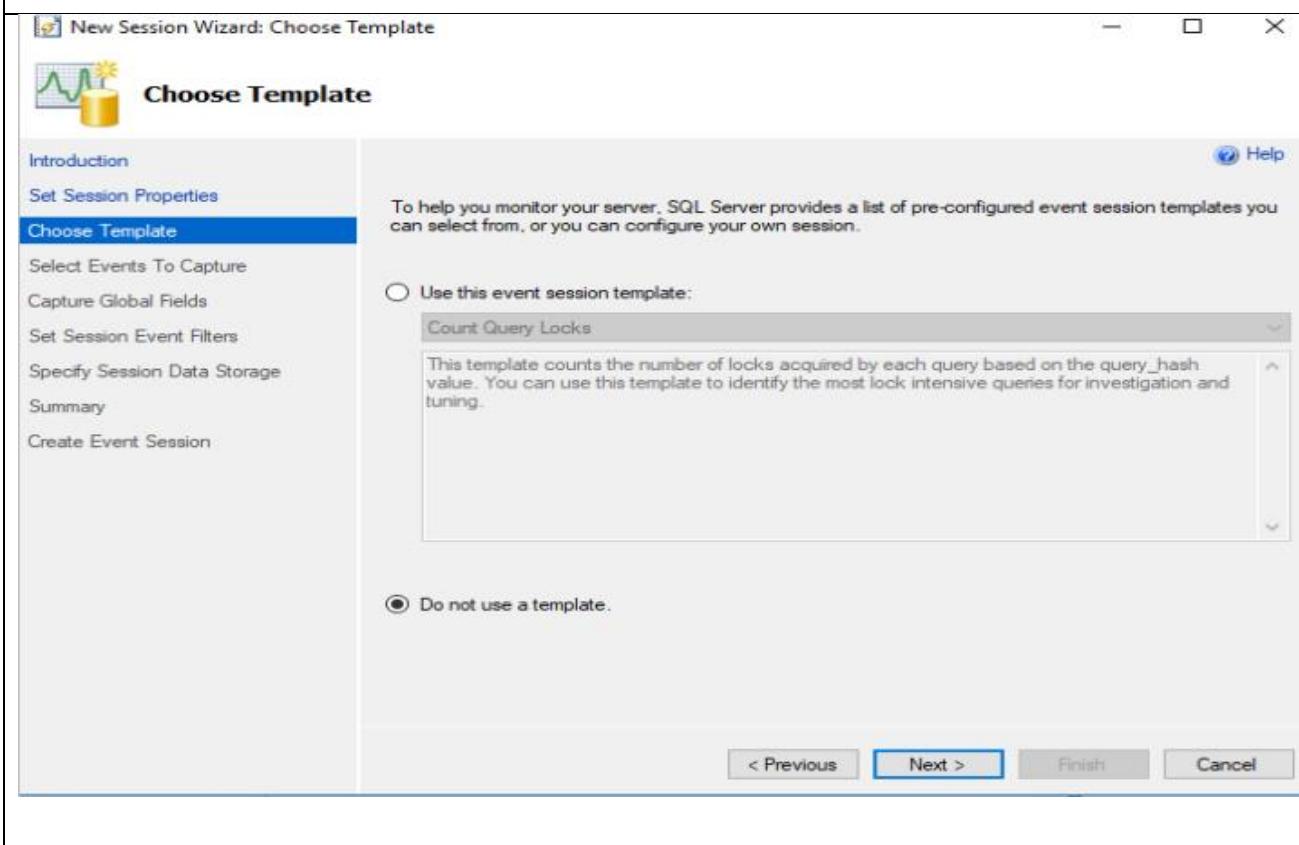
**Step 3:** After you select the new session wizard option the dialog box similar to below will appear which helps us in creating events for troubleshooting and performance analysis. Now select the next option to go to next option i.e. set session properties.



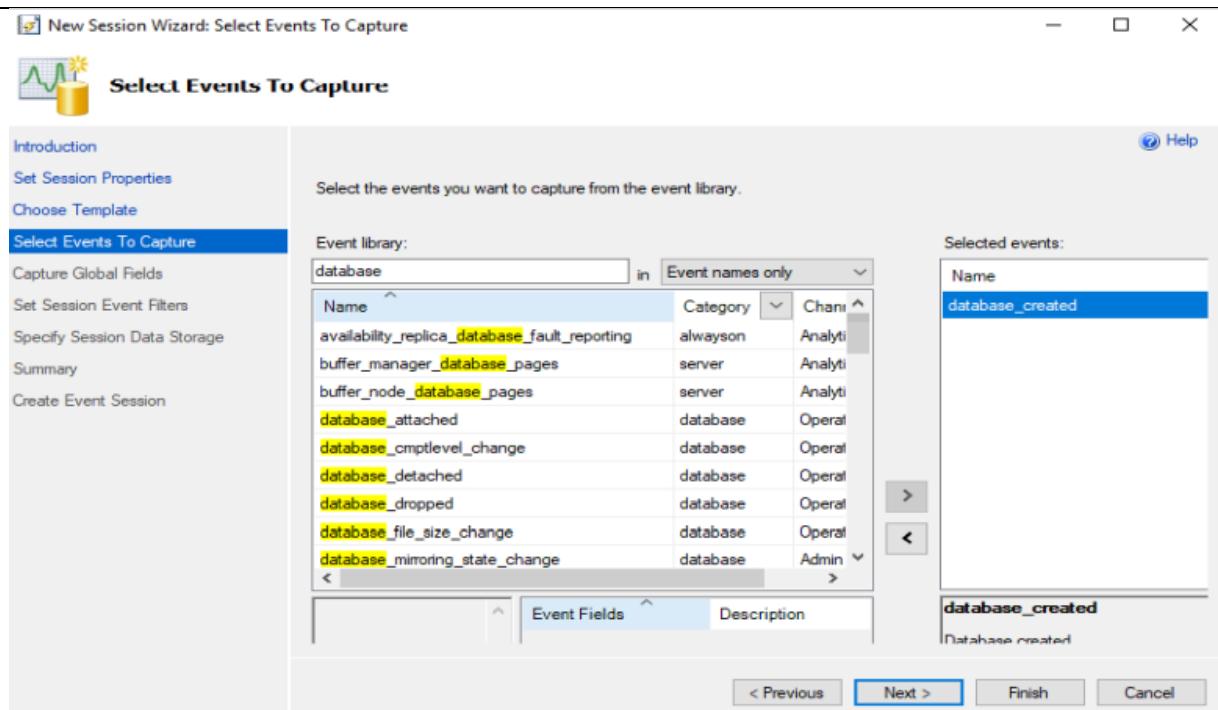
**Step 4:** In this dialog box we have to provide the session name for event. In my case I have named it as the system name i.e. LibraryManagement then select next.



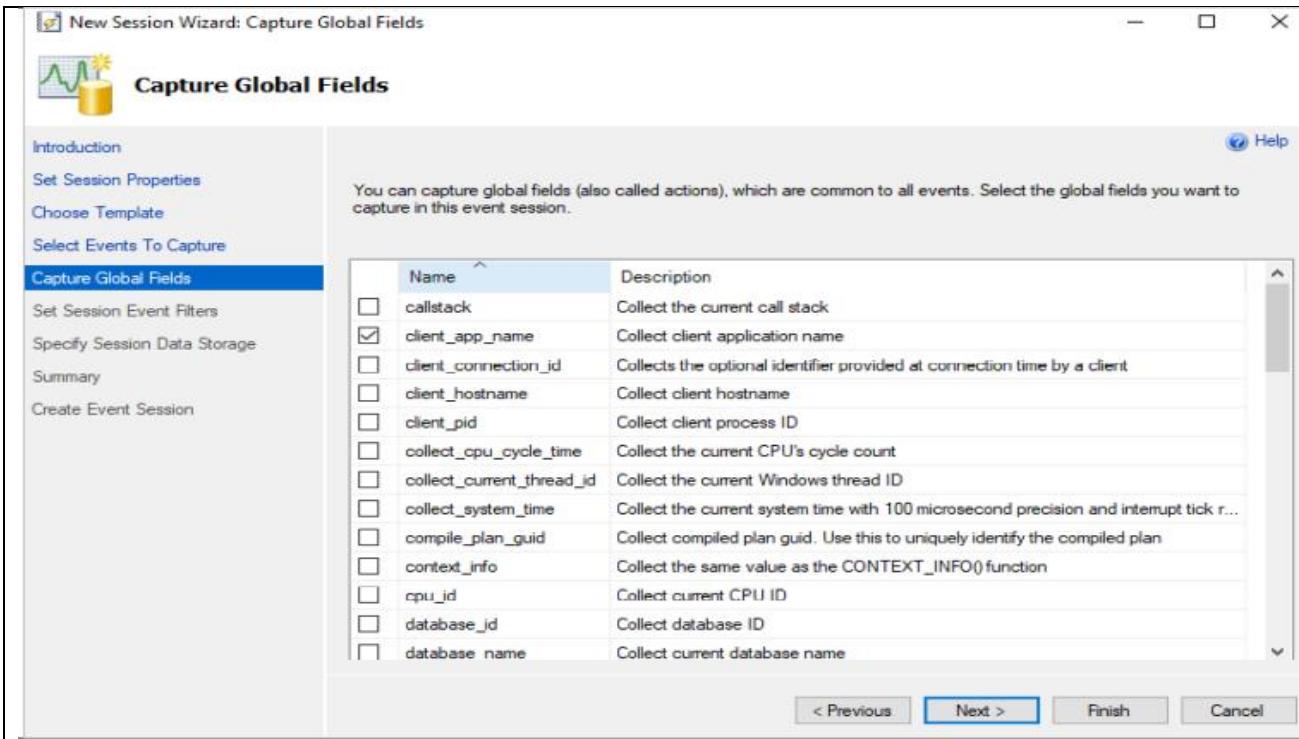
**Step 5:** As we aren't using any session template for the event click on the don't use template checkbox and then select next button.



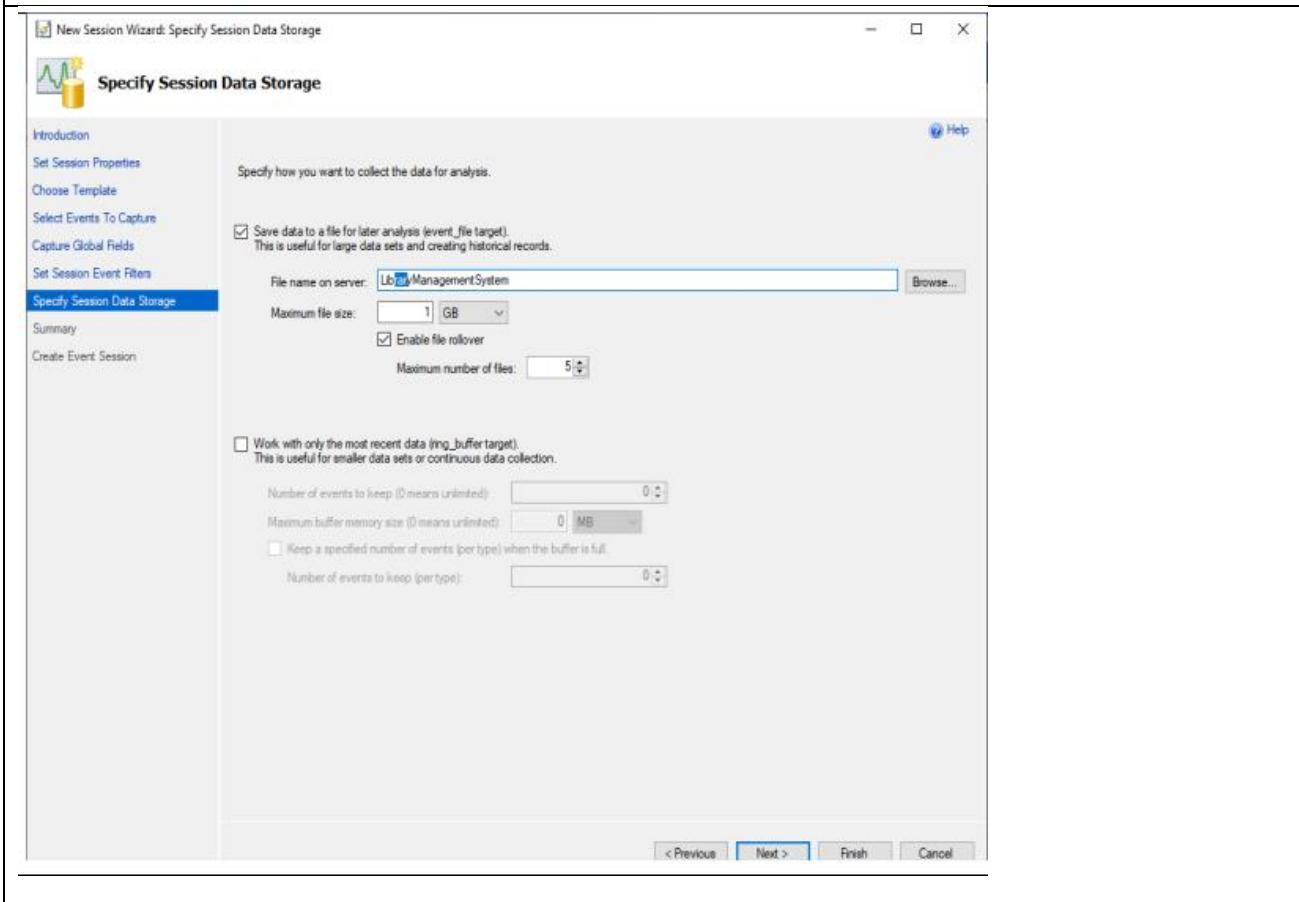
**Step 6:** Here we can select the event which we want to capture like if database is dropped or the database file changed. If the selected events occurs then we will be notified about it and if it is unwanted then we can take the required action in no time.



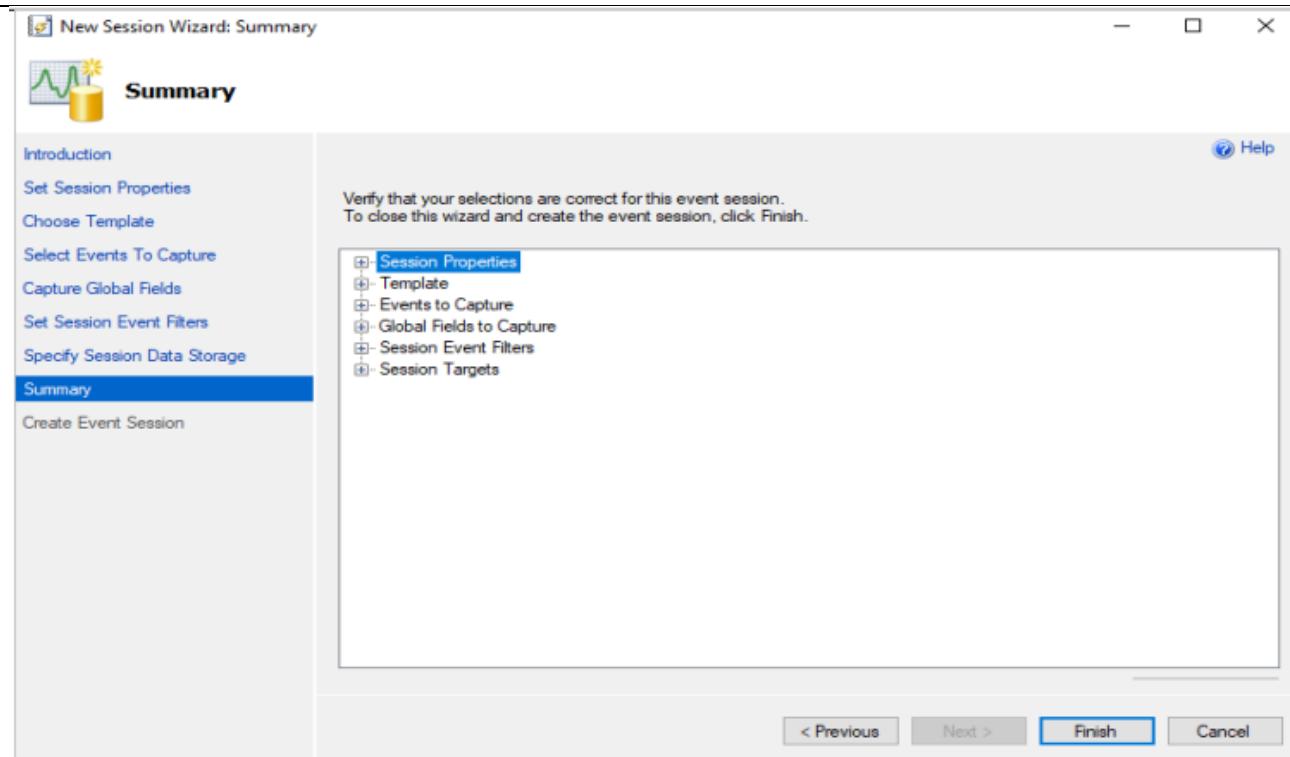
**Step 7:** Then in the capture global fields we can capture global field which are common to all the events. We can select the global fields which we want to capture in this event session.



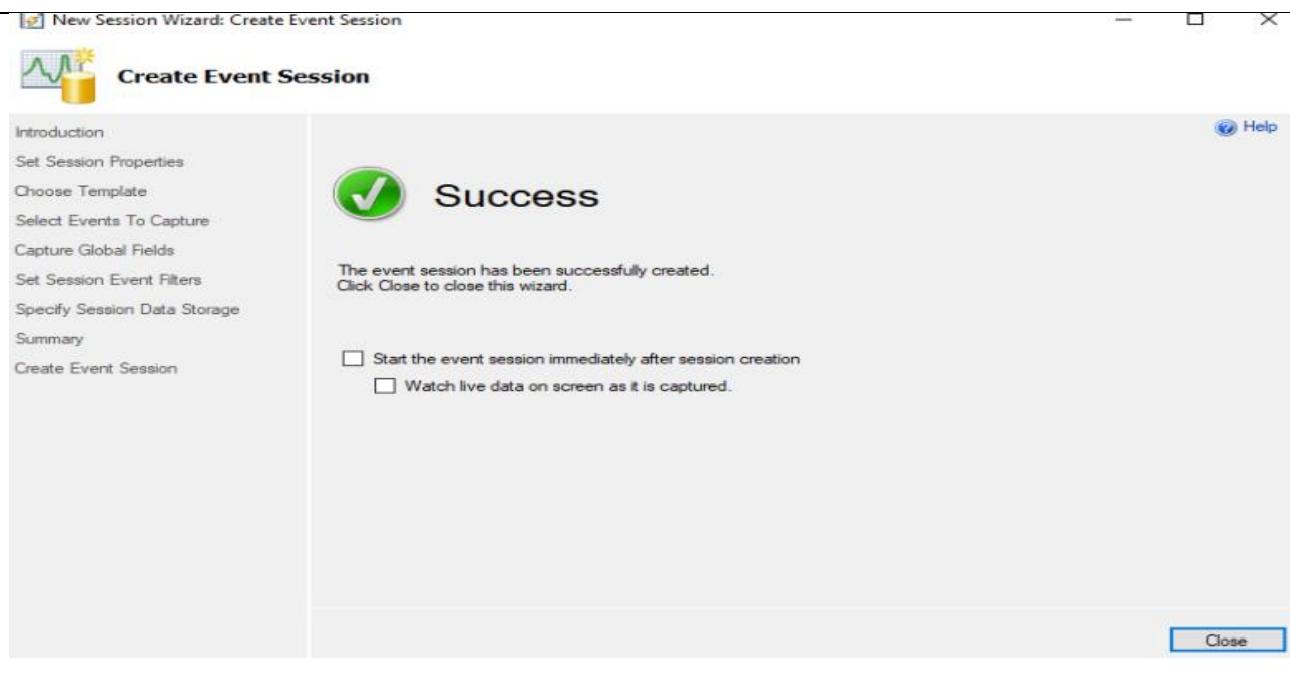
**Step 8:** If we want to analyze the events later or perform any action after seeing the events then we can also save data to a file for later analysis using this feature



Step 9: While setting the event wizard if we want to check if our selection are correct for this event session then we can perform this in this dialog box. If all the actions are correct then we can create the event after selecting finish option.



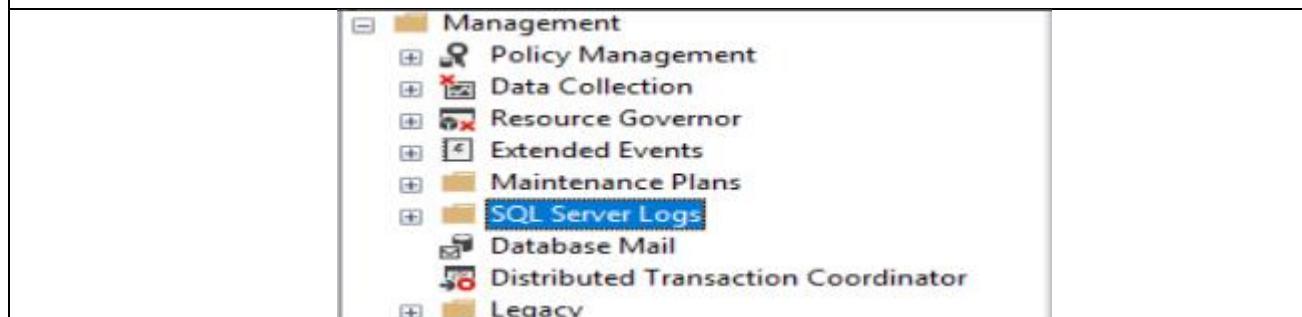
**Step 10:** After the event is successfully created below dialog box will appear. It's possible sometimes that some error might be encountered



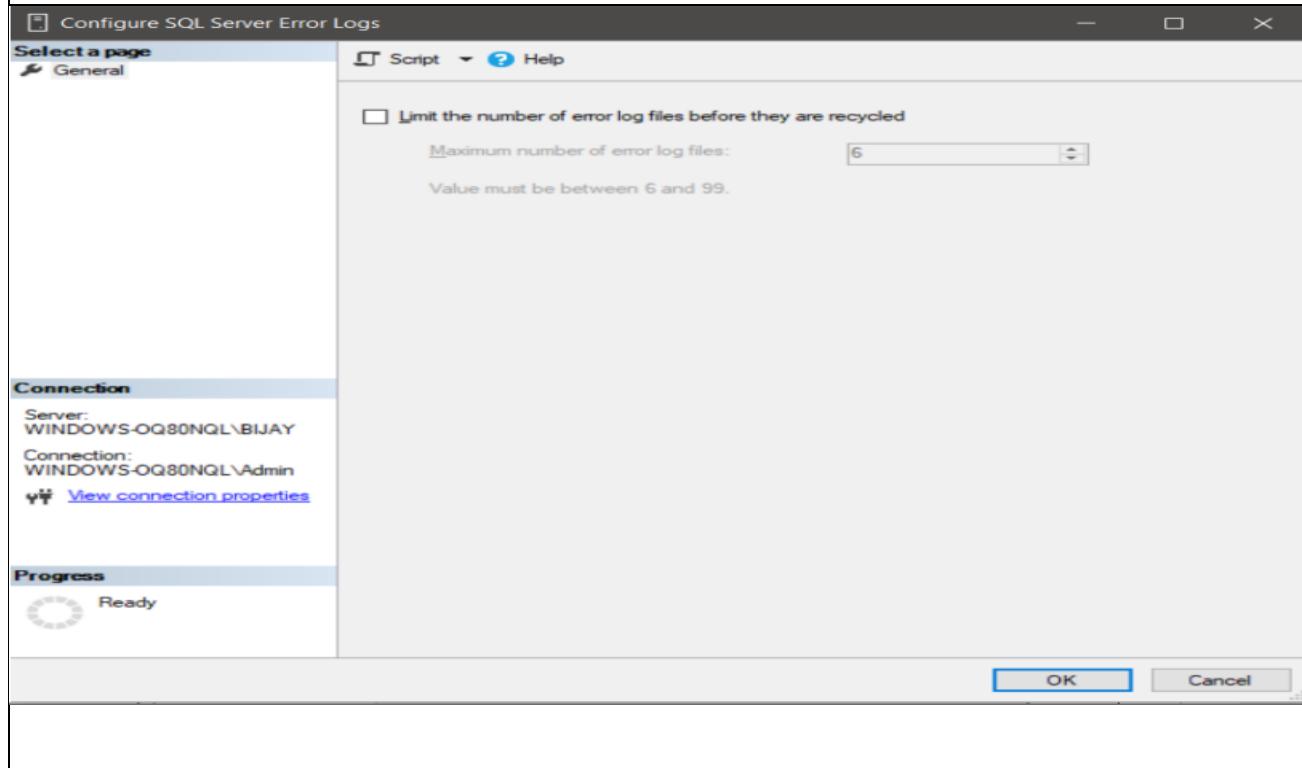
## Error Log

Error log is a record of critical errors that are encountered by the database server while in operation. Some of the common entries in an error log include the table corruption and configuration corruption and in many cases they are very useful tools for troubleshooting and managing the database servers. The error log is a central file where all important actions and error messages related to the SQL Server are recorded. The database administrator should examine the log every day. The following procedure describes how to create and view the error log with the Microsoft SQL Server Management Studio.

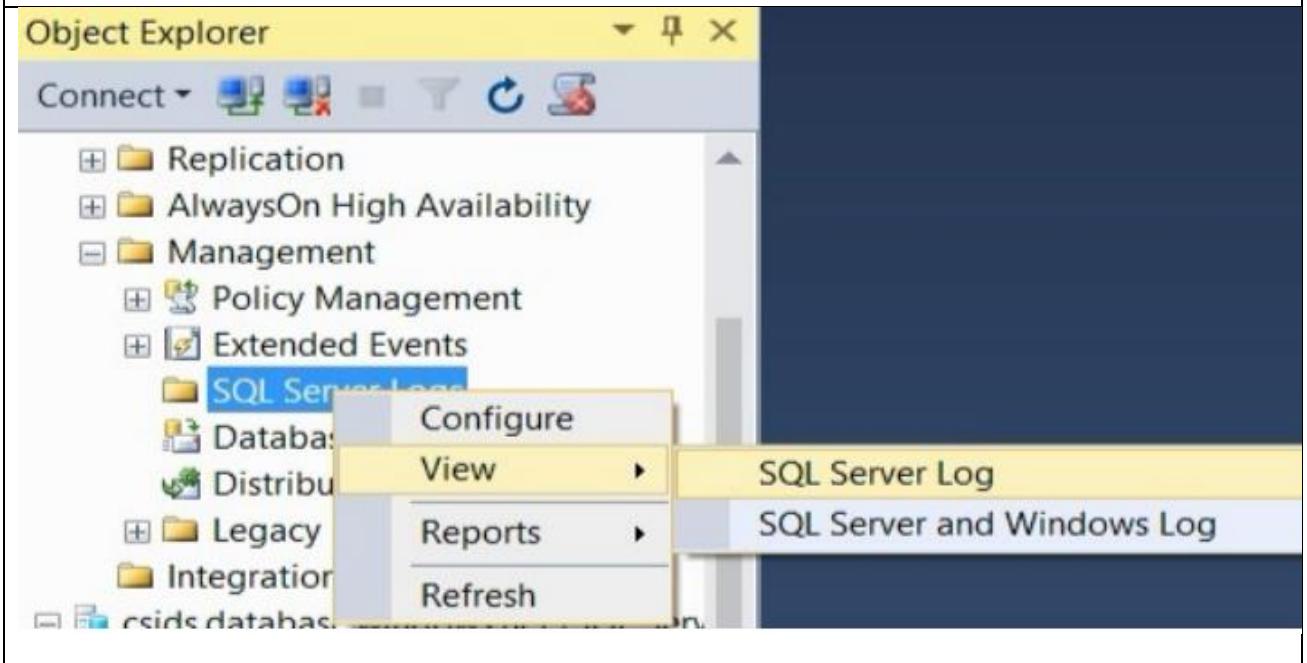
**Step 1:** After opening the Microsoft SQL Server at the left in object explorer expand the management option and there we can see the SQL Server Logs. Now right click on the server logs option and click on option menu.



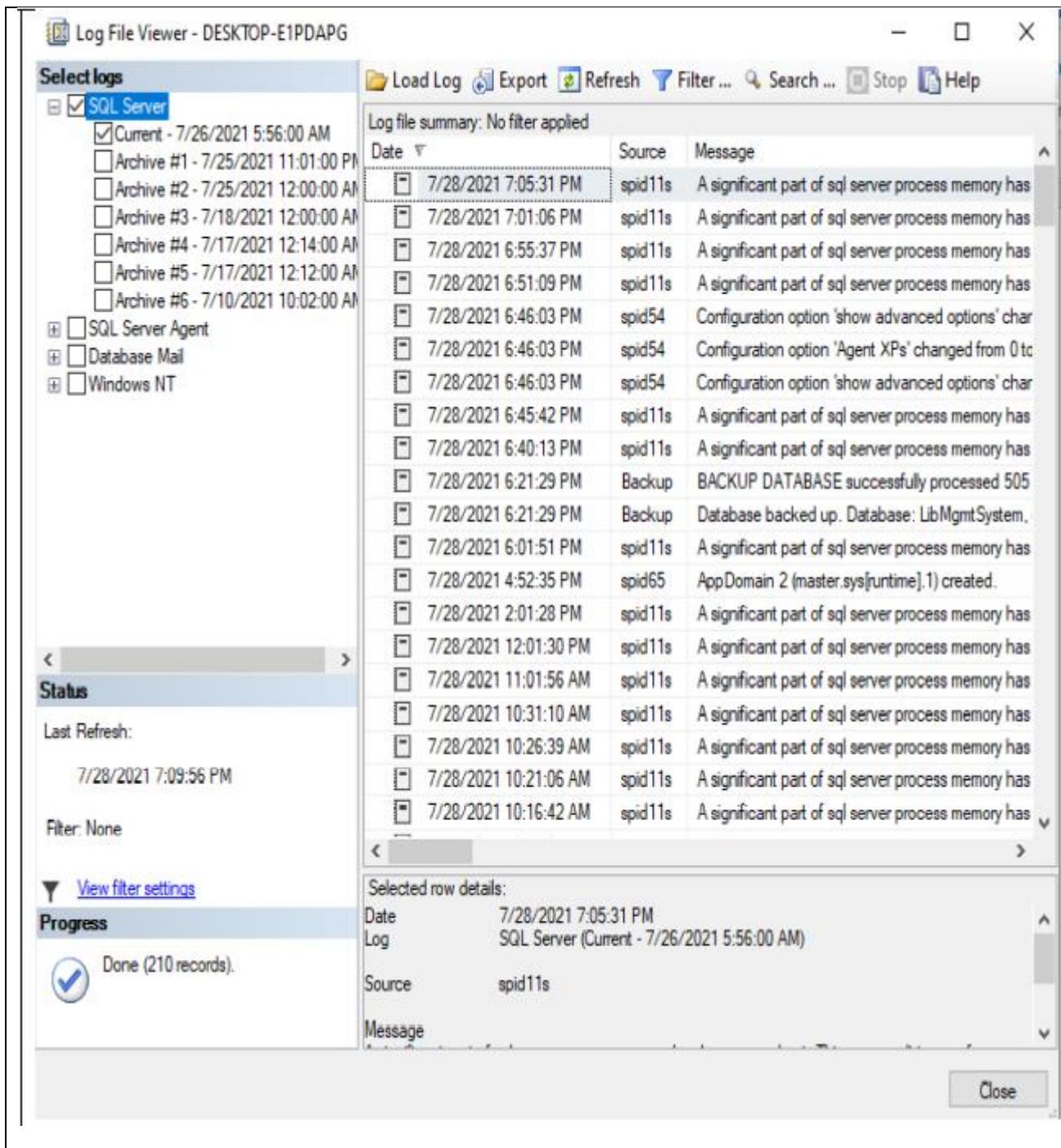
**Step 2:** Now click on the check box to limit number of log files and select maximum number of log files.



**Step 3:** Then after the configuration of error log expand the management option above and right click on sql server logs then select view and choose the sql server log.



**Step 4:** Now the log file will appear with the list of logs which we wanted to view.



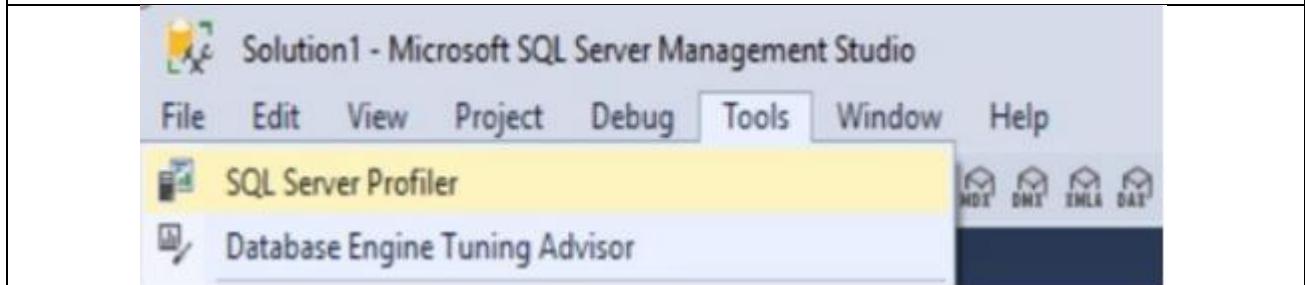
## SQL Server Profiler

“An SQL server profiler is a tool for tracing, recreating, and troubleshooting problems in MS SQL Server, Microsoft’s Relational Database Management System. It is an interface to create and manage traces and analyze and replay trace results. Events are saved in a trace file that can later be analyzed or used to replay a specific series of steps when trying to diagnose a problem.” (**GeeksforGeeks, 2020**)

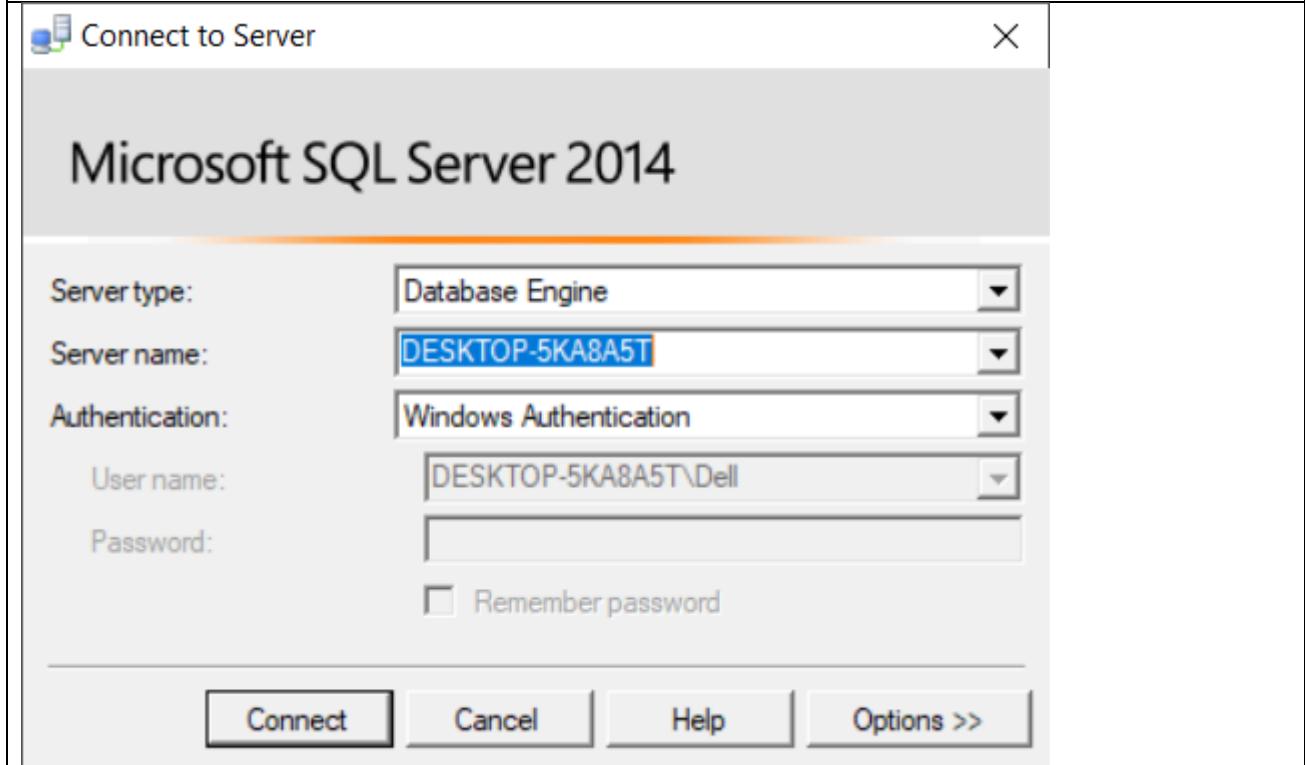
**For example**, we can monitor a production environment to see which stored procedures are affecting performance by executing too slowly. SQL Profiler may be used to capture data for real-time or future review. In addition, traces may be scheduled to run at specific times. For example, to compare

performance when the database is busy or idle, one trace may be set up to run in the morning at 10 a.m. when the database is busiest, and another set up to run at 2 a.m. when there is little or no database activity. Below I have provided the configuration screenshots of SQL Server profiler in MS SQL Server 2014.

**Step 1:** After opening the SQL server in the top navigation bar surf to tool menu and select SQL Server Profiler.

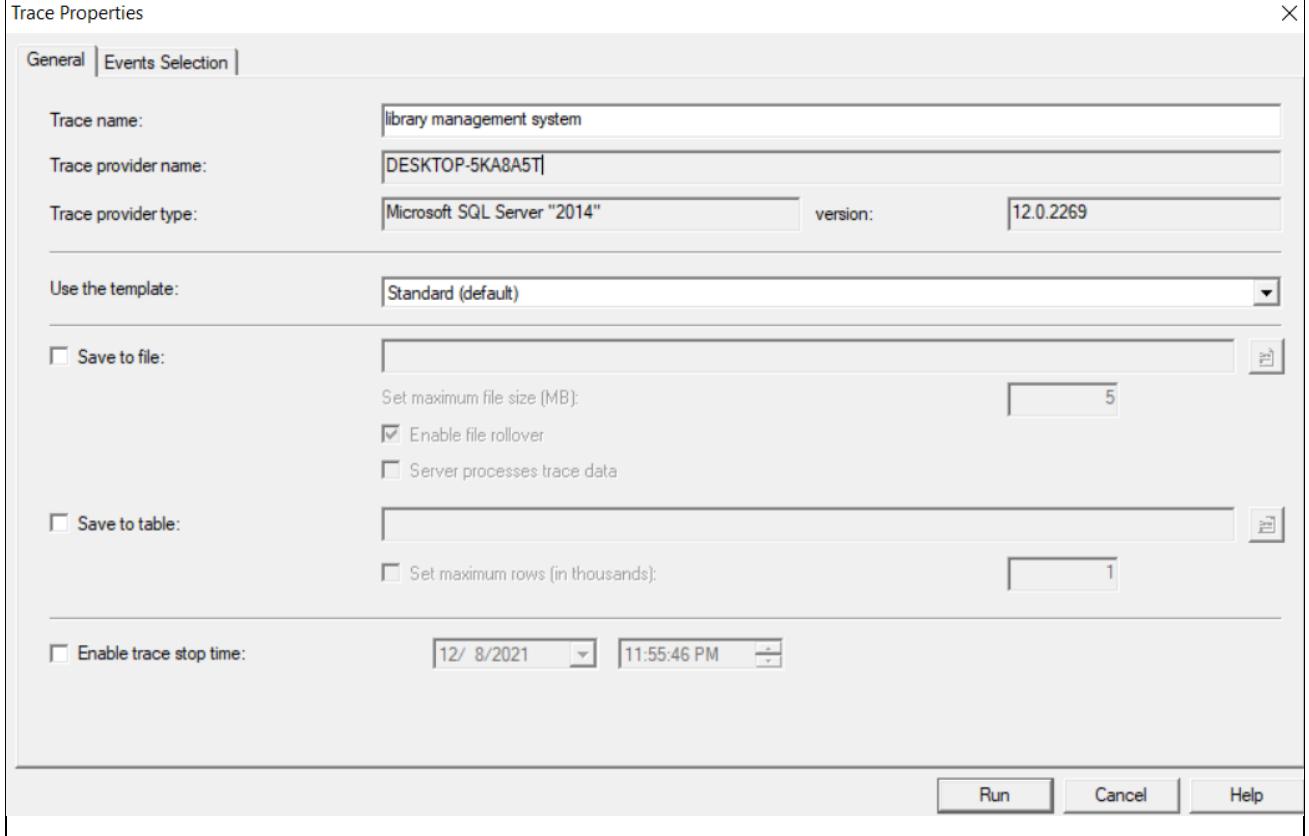


**Step 2:** Then SQL studio will open another window with the connecting dialog box. Click on the connect option and another window will open



**Step 3:** After this dialog box appears, click on the trace name and provide the desired name for the

trace properties and then select Run button for further configuration.



Step 4: Now right click the process that we want to profile and then click trace process in SQL Profiler

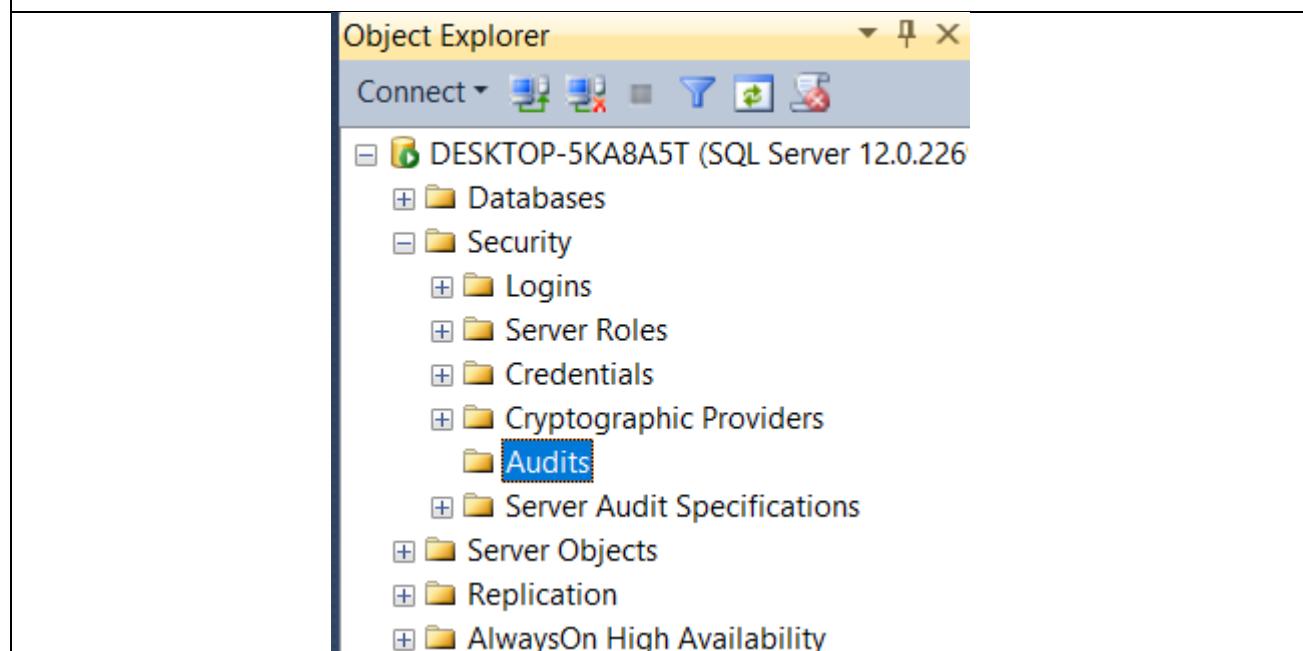
EventClass	TextData	ApplicationName	NTUserName	LoginName	CPU
Trace Start					
ExistingConnection	-- network protocol: LPC set quoted...	Microsoft SQ...	kusha	DESKTO...	
ExistingConnection	-- network protocol: LPC set quoted...	SQLServerCEIP		SQLTELE...	
ExistingConnection	-- network protocol: LPC set quoted...	SQLAgent - G...		SQLSERV...	NT SER...
ExistingConnection	-- network protocol: LPC set quoted...	SQLAgent - E...		SQLSERV...	NT SER...
ExistingConnection	-- network protocol: LPC set quoted...	Microsoft SQ...	kusha	DESKTO...	
ExistingConnection	-- network protocol: LPC set quoted...	Microsoft SQ...	kusha	DESKTO...	
ExistingConnection	-- network protocol: LPC set quoted...	Microsoft SQ...	kusha	DESKTO...	

```
-- network protocol: LPC
set quoted_identifier off
set arithabort off
set numeric_roundabort off
set ansi_warnings on
set ansi_padding on
set ansi_nulls on
set concat_null_yields_null on
set cursor_close_on_commit off
set implicit_transactions off
<
Trace is running.
```

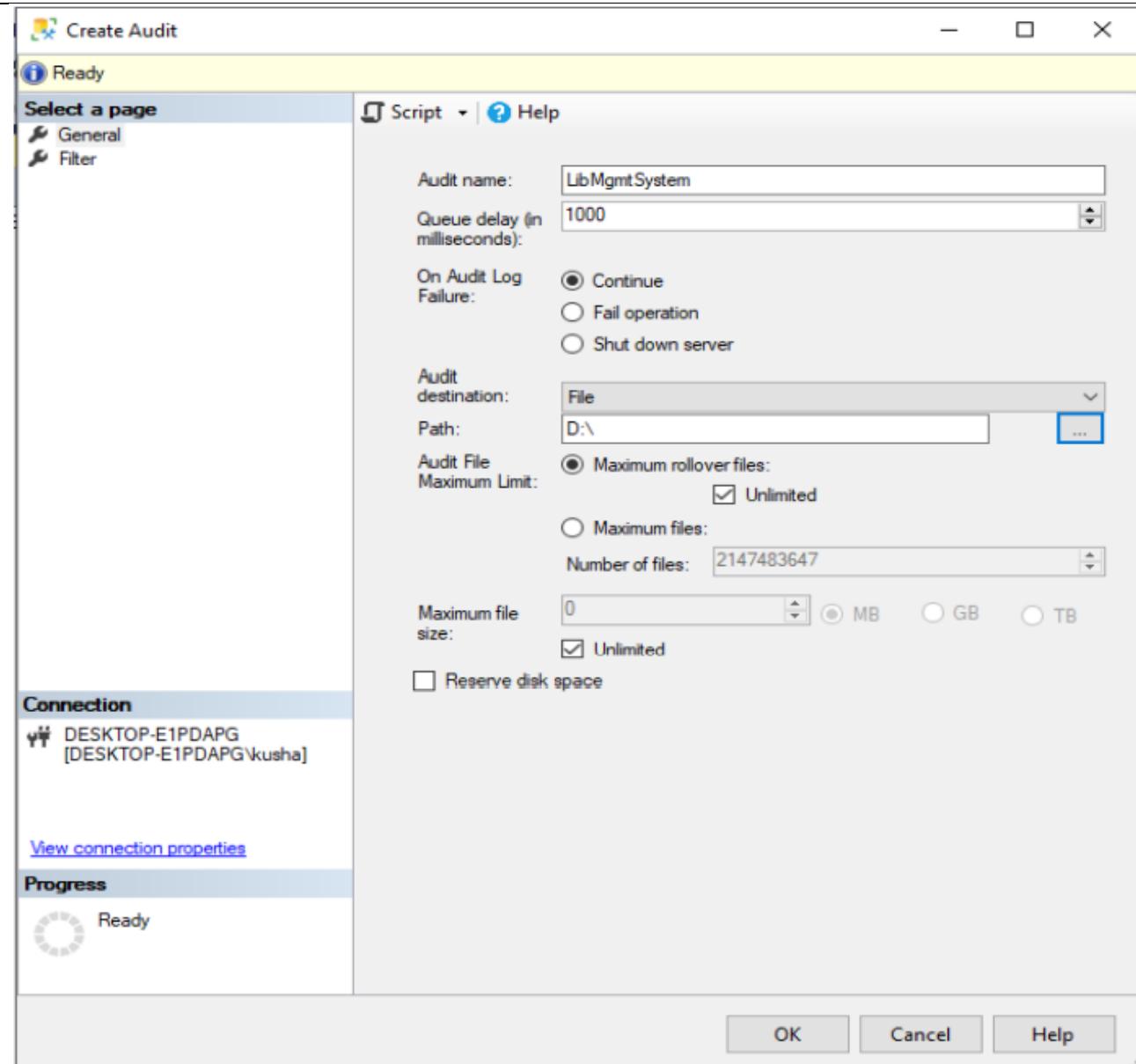
## Audit Logs

SQL Server auditing is a new feature which makes use of extended events to allow you to audit everything that happens in your server, from server setting changes all the way down to who modified a value in a specific table in the database. This information is then written the Windows security log, the Windows application log or to a flat file. Below I have provided the steps to perform audit logs in SQL Server 2014;

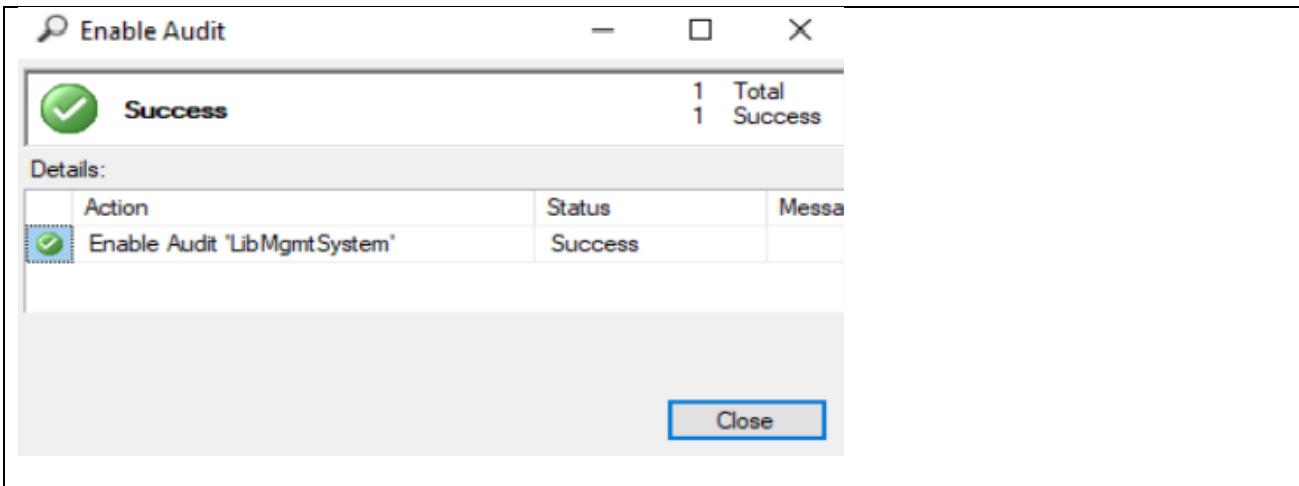
**Step 1:** Open your database software and then in the object explorer then right click on the audit option and click new audit.



**Step 2:** After selecting create audit this dialog box will appear. Here provide the audit name and file path as both are necessary. For other options we can perform as required for our system and then click ok.



**Step 3:** After filling the desired data in previous dialog box and clicking ok the audit will be enabled.



## Activity Monitor

Activity Monitor runs queries on the monitored instance to obtain information for the Activity Monitor display panes. When the refresh interval is set to less than 10 seconds, the time that is used to run these queries can affect server performance. To view actual activity, you must have VIEW SERVER STATE permission. To view the Data File I/O section of Activity Monitor, you must have CREATE DATABASE, ALTER ANY DATABASE, or VIEW ANY DEFINITION permission in addition to VIEW SERVER STATE.

<p>Step 1: Right-click on the item explorer and select Activity Monitor from the drop-down menu.</p>	<p>The screenshot shows the 'Object Explorer' window in SQL Server Management Studio. A context menu is open over a database node. The menu items include 'Connect...', 'Disconnect', 'Register...', 'New Query', 'Activity Monitor' (which is highlighted with a yellow background), 'Start', 'Stop', 'Pause', 'Resume', 'Restart', 'Policies', 'Facets', 'Start PowerShell', 'Reports', 'Refresh', and 'Properties'. The 'Activity Monitor' option is the second item in the list.</p>
--	--

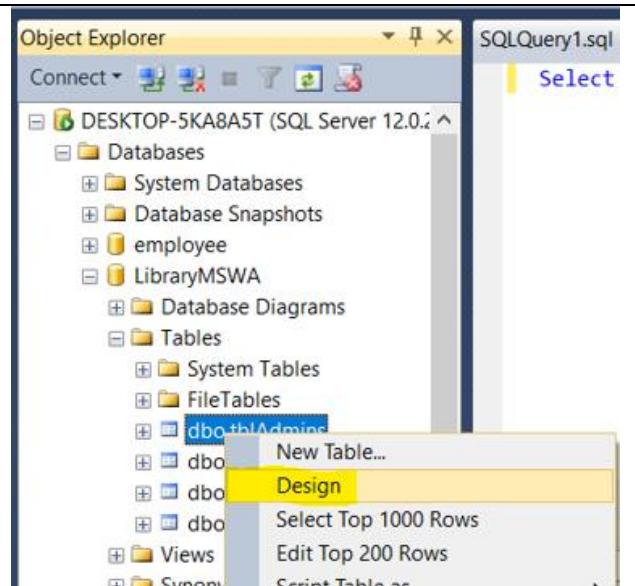
<p><b>Step 2:</b> The Activity Monitor window's overview will display.</p>	<p><b>Processes</b></p> <p><b>Resource Waits</b></p> <p><b>Data File I/O</b></p> <p><b>Recent Expensive Queries</b></p> <p><b>Active Expensive Queries</b></p>																																																							
<p><b>Step 3:</b> The resources Waits option displays information on resource waits such as wait category, wait time, and other factors. Recent Expensive Queries shows information on recent/active queries</p>	<p><b>Resource Waits</b></p> <table border="1"> <thead> <tr> <th>Wait Category</th> <th>Wait Time (ms/sec)</th> <th>Recent Wait Time...</th> <th>Average Waiter C...</th> <th>Cumulative Wait Tim...</th> </tr> </thead> <tbody> <tr> <td>Buffer I/O</td> <td>0</td> <td>0</td> <td>0.0</td> <td>19</td> </tr> <tr> <td>Buffer Latch</td> <td>0</td> <td>0</td> <td>0.0</td> <td>0</td> </tr> <tr> <td>Latch</td> <td>0</td> <td>0</td> <td>0.0</td> <td>0</td> </tr> <tr> <td>Lock</td> <td>0</td> <td>0</td> <td>0.0</td> <td>9</td> </tr> <tr> <td>Logging</td> <td>0</td> <td>0</td> <td>0.0</td> <td>0</td> </tr> <tr> <td>Memory</td> <td>0</td> <td>0</td> <td>0.0</td> <td>0</td> </tr> <tr> <td>Network I/O</td> <td>0</td> <td>0</td> <td>0.0</td> <td>0</td> </tr> <tr> <td>Other</td> <td>0</td> <td>0</td> <td>0.0</td> <td>0</td> </tr> </tbody> </table> <p><b>Data File I/O</b></p> <table border="1"> <thead> <tr> <th>Database</th> <th>File Name</th> <th>MB/sec Read</th> <th>MB/sec Wri...</th> <th>Response Tim...</th> </tr> </thead> <tbody> <tr> <td>Library Man...</td> <td>C:\Program Files\Microsoft SQL Server\MSSQL15 AN...</td> <td>0.0</td> <td>0.0</td> <td>0</td> </tr> </tbody> </table>	Wait Category	Wait Time (ms/sec)	Recent Wait Time...	Average Waiter C...	Cumulative Wait Tim...	Buffer I/O	0	0	0.0	19	Buffer Latch	0	0	0.0	0	Latch	0	0	0.0	0	Lock	0	0	0.0	9	Logging	0	0	0.0	0	Memory	0	0	0.0	0	Network I/O	0	0	0.0	0	Other	0	0	0.0	0	Database	File Name	MB/sec Read	MB/sec Wri...	Response Tim...	Library Man...	C:\Program Files\Microsoft SQL Server\MSSQL15 AN...	0.0	0.0	0
Wait Category	Wait Time (ms/sec)	Recent Wait Time...	Average Waiter C...	Cumulative Wait Tim...																																																				
Buffer I/O	0	0	0.0	19																																																				
Buffer Latch	0	0	0.0	0																																																				
Latch	0	0	0.0	0																																																				
Lock	0	0	0.0	9																																																				
Logging	0	0	0.0	0																																																				
Memory	0	0	0.0	0																																																				
Network I/O	0	0	0.0	0																																																				
Other	0	0	0.0	0																																																				
Database	File Name	MB/sec Read	MB/sec Wri...	Response Tim...																																																				
Library Man...	C:\Program Files\Microsoft SQL Server\MSSQL15 AN...	0.0	0.0	0																																																				

that require resources or CPU execution time, and Data File I/O displays the I/O file.

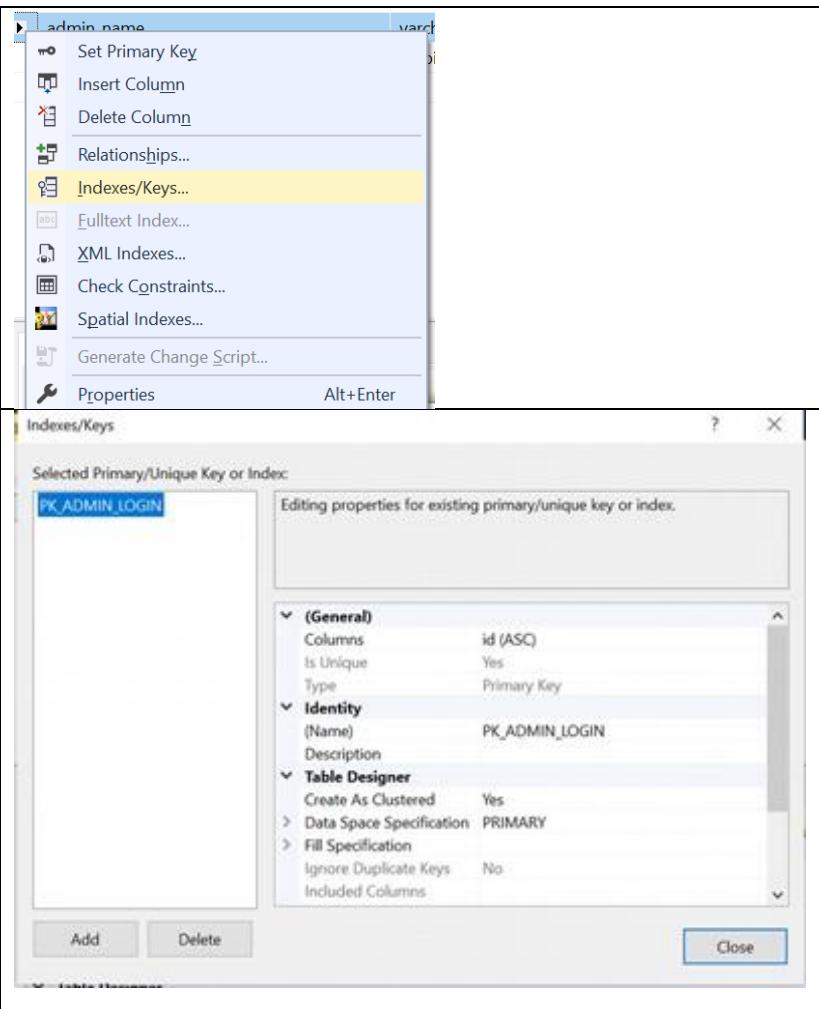
### **Indexes/Key:**

An index is a set of keys made up of one or more columns in a table or view. These keys are stored in a structure (B-tree) that allows SQL Server to easily and efficiently locate the row or rows associated with the key values. When PRIMARY KEY and UNIQUE constraints are established on table columns, indexes are generated automatically. Here, I will demonstrate the Table indexing process. I will use SQL management studio 2019.

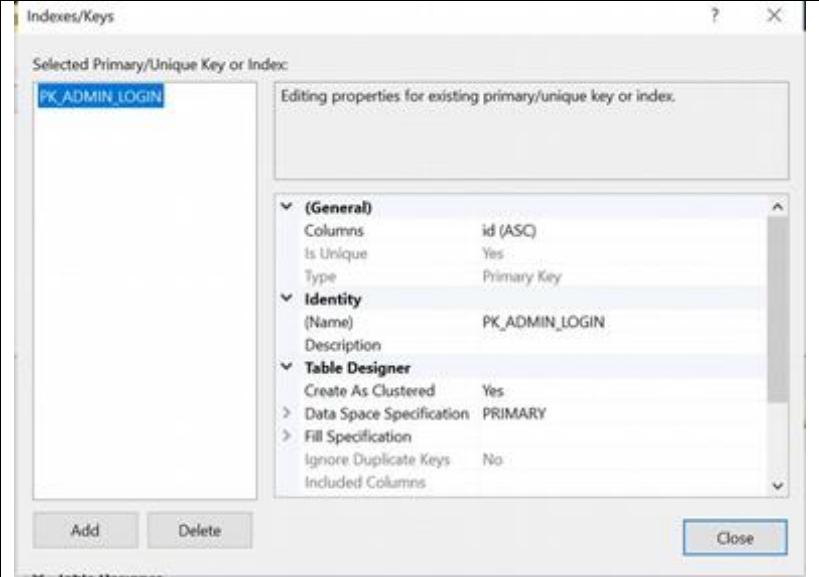
Step 1. Right click on your table and click on design:



Step 2. Click on special column where you want to add indexing and press on indexes/key option

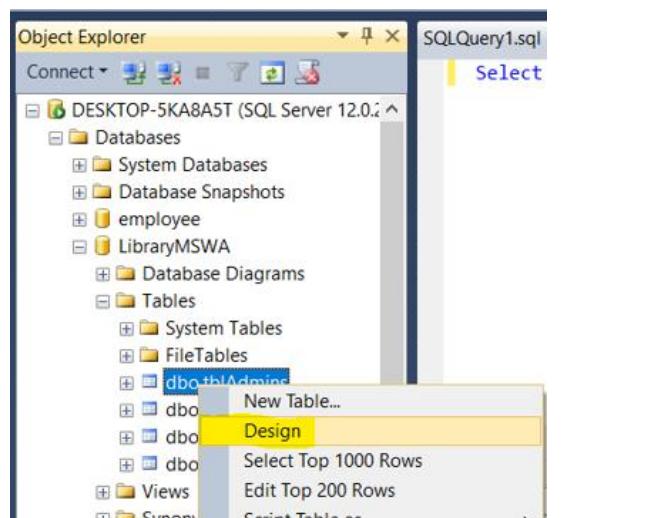


Step 3. After that you can add special indexes and click on ADD button after that you have been success to add indexes on your table

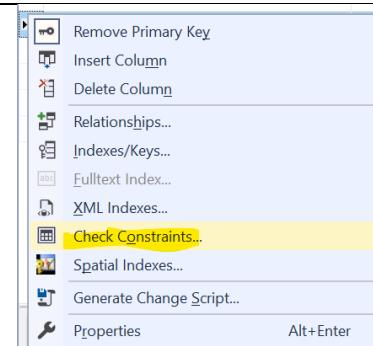


**Constraints:** Constraints are a useful way to improve query efficiency while still protecting data confidentiality, but there is still a trade-off: The data takes up more space, and changes take longer and are more complex. Constraints are an important feature that can dramatically boost SQL query performance. Here, I will demonstrate the SQL constraints on SQL management server 2019.

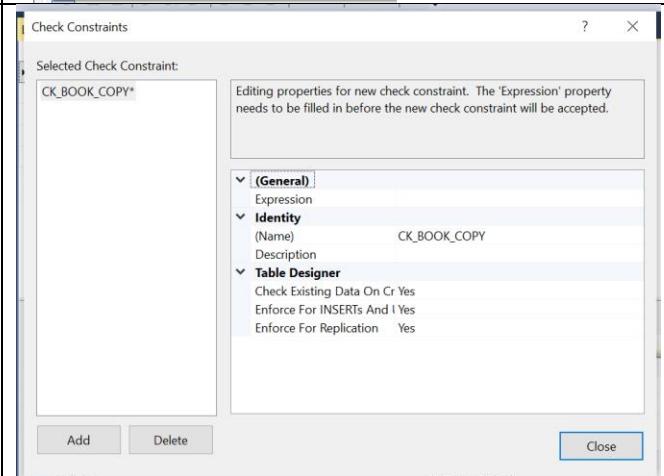
Step 1. Right click on you table and click on design



Step 2. Click on special column where you want to add indexing and press on Constraint option.



Step 3. After that, you can add special constraint for your attributes.



**Trigger:** A trigger is a database stored operation that is automatically called when a certain occurrence happens in the database. When a row is added into a certain table or some table columns are modified, for example, a button may be triggered.

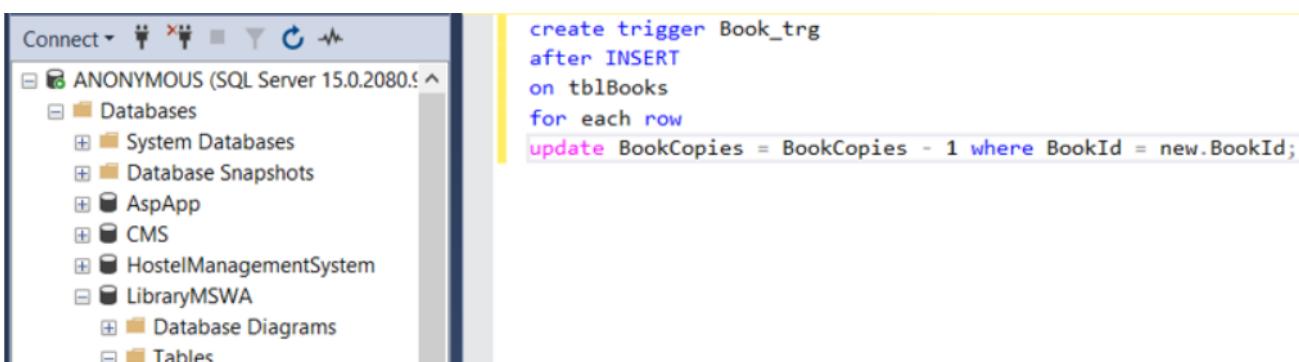


Figure: Trigger Processor

With the help of database trigger we can also improve the database performance as per requirement. A trigger is a type of stored process that runs automatically when a database server event occurs.

### Tools available in the system to manage security and authorizations

**Security:** The term "information protection" refers to a collection of tools, controls, and procedures for ensuring and maintaining database security, integrity, and availability. Since security is the aspect that

is breached in the majority of data breaches, this part would concentrate on it. Database security must address and protect the following:

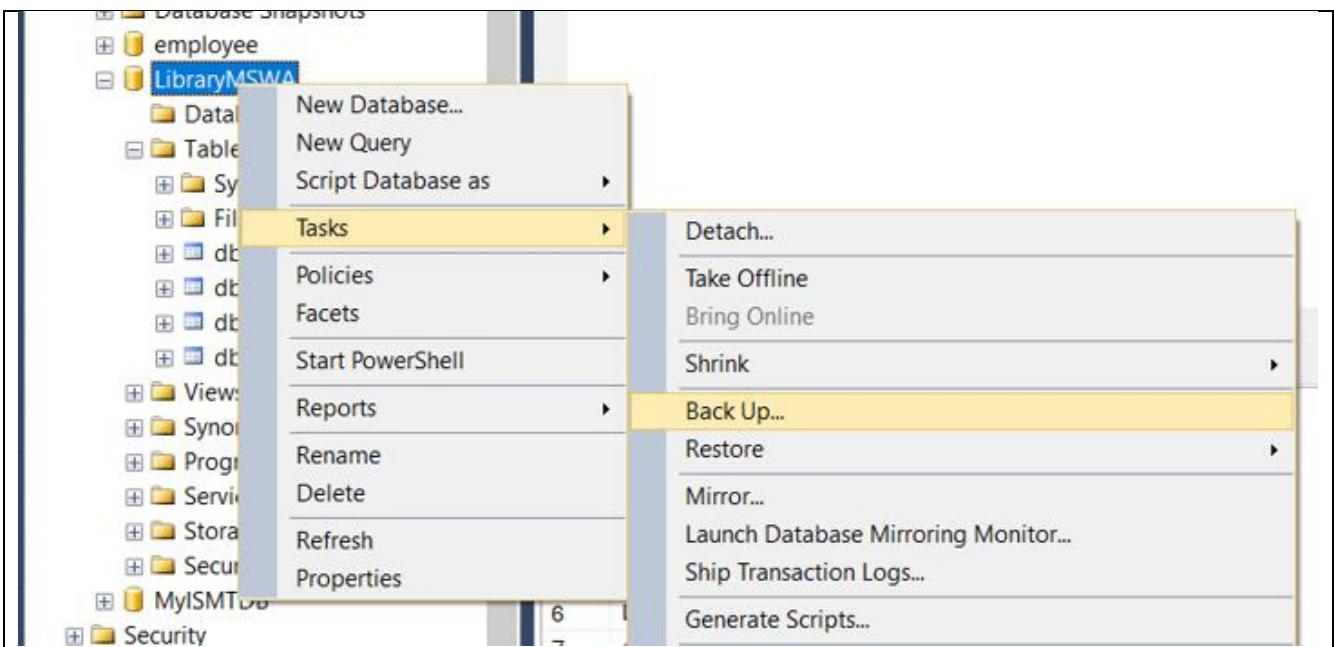
- The data in the database
- The database management system (DBMS)
- Any associated applications
- The physical database server and/or the virtual database server and the underlying hardware
- The computing and/or network infrastructure used to access the database

Database security is a complicated and difficult task that encompasses all facets of information technology systems and procedures. It's also incompatible with database usability. The database becomes more vulnerable to security attacks as it becomes more available and usable; the database becomes more invulnerable to threats as it becomes more difficult to navigate and use.

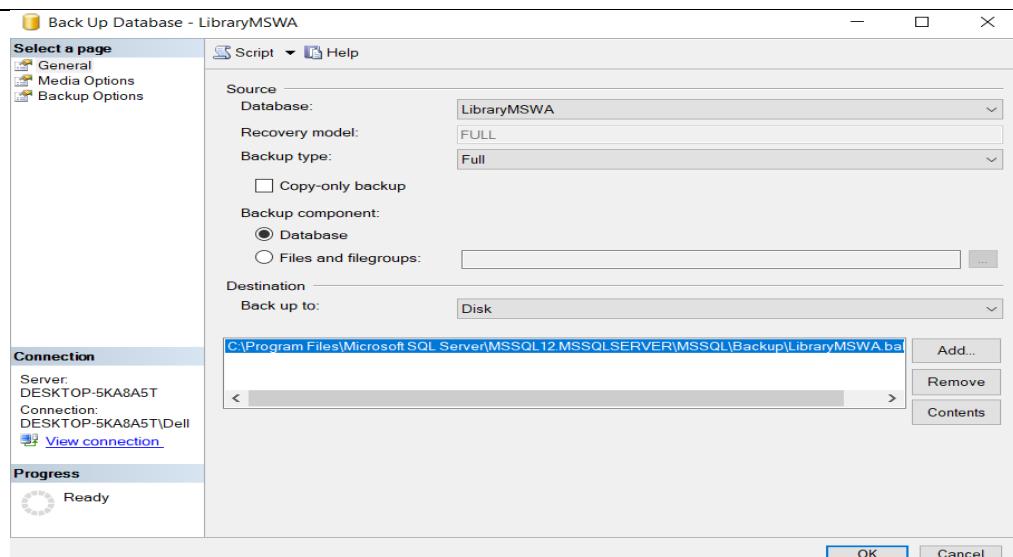
### **Database Backup:**

A backup, also known as a data backup, is a copy of computer data that is taken and saved elsewhere so that it can be used to retrieve the original in the event of a data failure event. The method of backing up the operating state, design, and storage data of database applications is known as database backup. In the event that the main database fails, is compromised, or is destroyed, it allows the existence of a replicated instance or replica of the database. Here, I will show the process of database on our system.

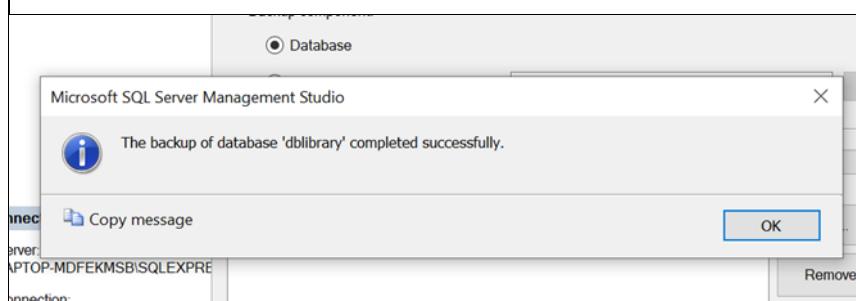
step 1 Open LMS and go to your specific database folder and click the right side and open Task after that open the backup option.



**Step 2:** Select the special path for backup file and click on Ok button



**Step 3:** Now your database backup has been successful.



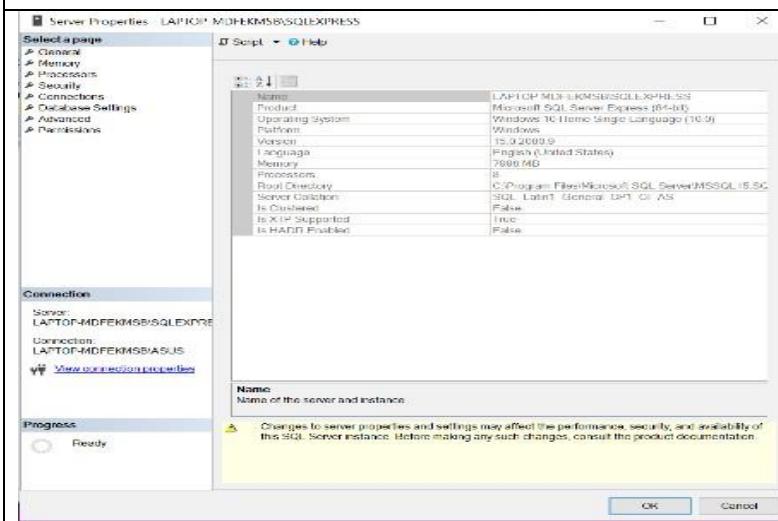
## Advantages of Database Backup:

- Our recovery solution is simple to set up and maintain, and it will restore a missing file from as far back as 30 days, meaning you don't have to think about losing the important file if you accidentally remove it and clear the recycle bin.
- You can restore the database to the point-in-time if suddenly there is an error occurs in the database A full database backup provides a full safety against data loss.
- The full database backup backs up the details of every transaction so that can be restored at the time of recovery.

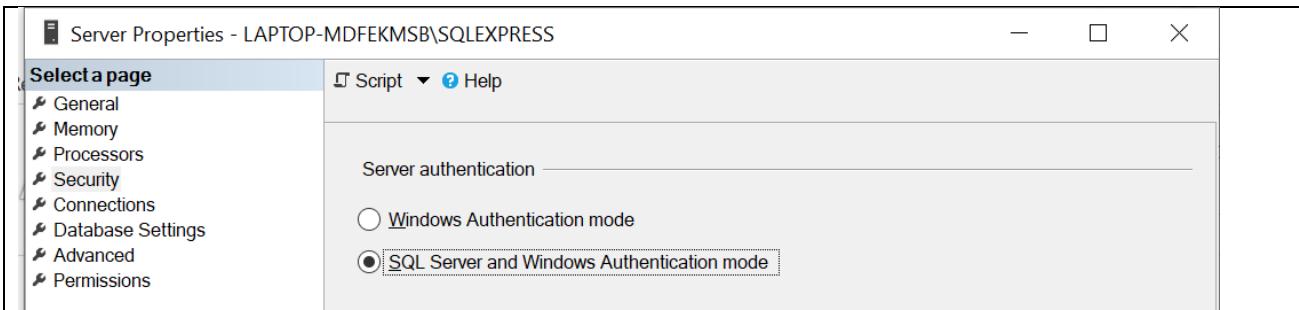
## System Authorizations:

System authorization, which is achieved by the validation and credentialing process, is a standardized, written approval that a system has proper security safeguards before it goes live. In Studio modes, the Authorization System allows developers and specific users to monitor access to LMS projects. This is accomplished by granting special privileges to users and user classes, as well as setting access keys to different uniPaaS project components. To monitor what each user can and cannot do in LMS, uniPaaS provides a modular authorization scheme. The permission scheme allows the program creator or system supervisor to restrict access to particular tasks in uniPaaS to users who have been officially approved. Here, I will show the process of system authorization.

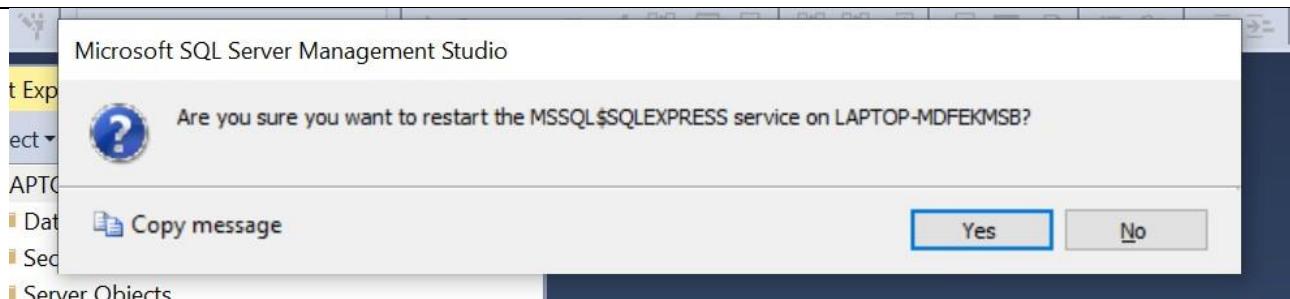
Step 1. Go to LMS and right click on your server name and open properties.



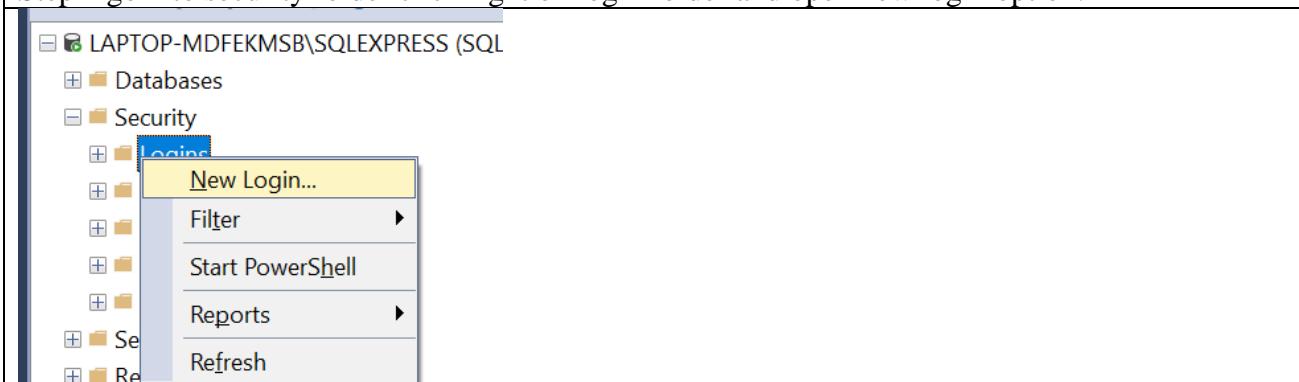
Step 2. Go into security option and choose SQL server and window authentication mode and click ok



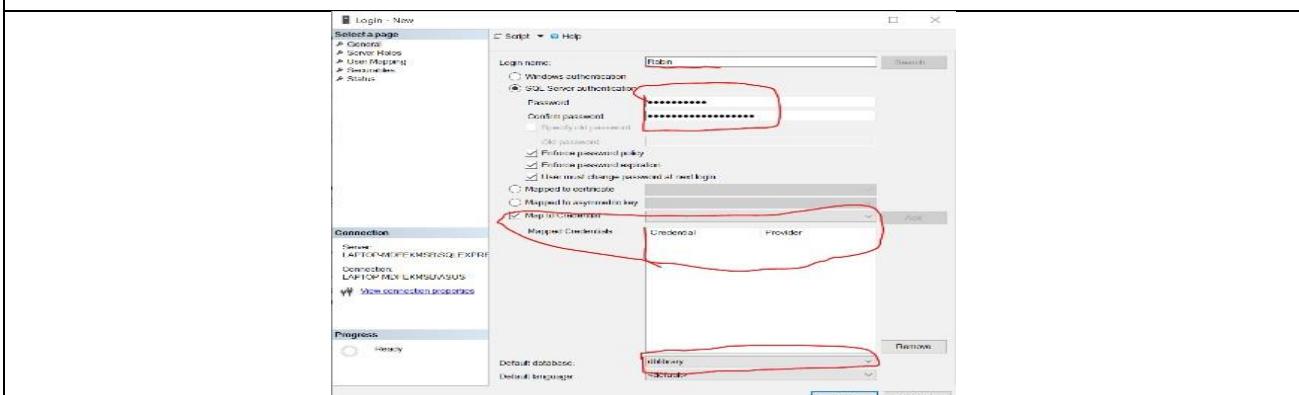
### Step 3 Restart your server



### Step 4 go into security folder click right on login folder and open new login option.



Step 5 choose SQL Server authinaction and enter server name, password and confirm password and give the specific database access and press on ok button.



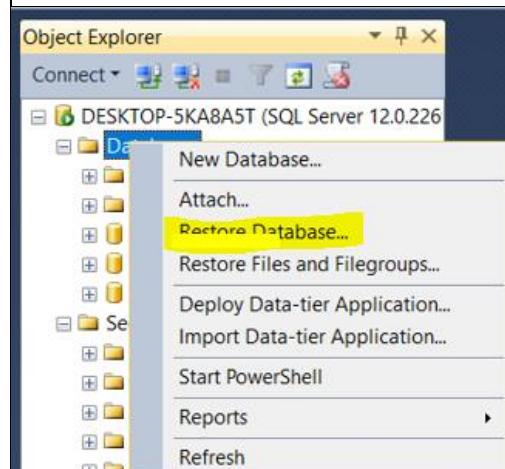
### Benefits of system authorization:

- Authorization lists make it easier to keep track of who has what authority. The permission list's user authority is established, not the actual items on the list.
- Giving a user control over all of the items on the list can be done with only one process.
- The lot of individual authority on the device is reduced by using authorization lists. The permission list is the only item that each user has private control over.
- Authorization lists provide a way to remember authorities when an object is saved. When an object is saved that is secured by an authorization list, the name of the authorization list is saved with the object.
- When saving the device (SAVSYS) or saving the protection details, the performance is improved (SAVSECDTA).

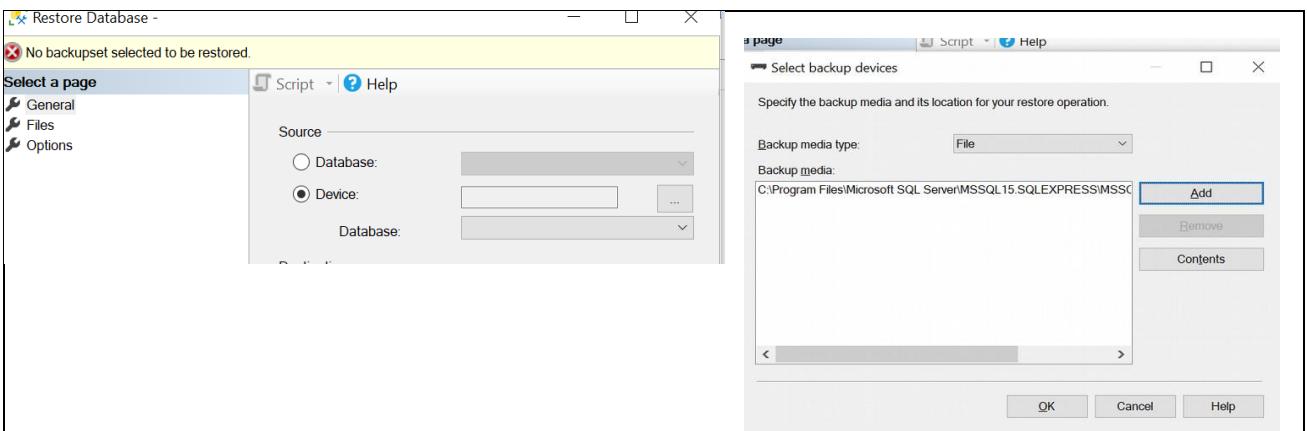
### Restore Database:

We can easily restore our backup file on our system. After backup we can easily developed database with backup file. Here, I will show the process of restore database on SQL server management studio.

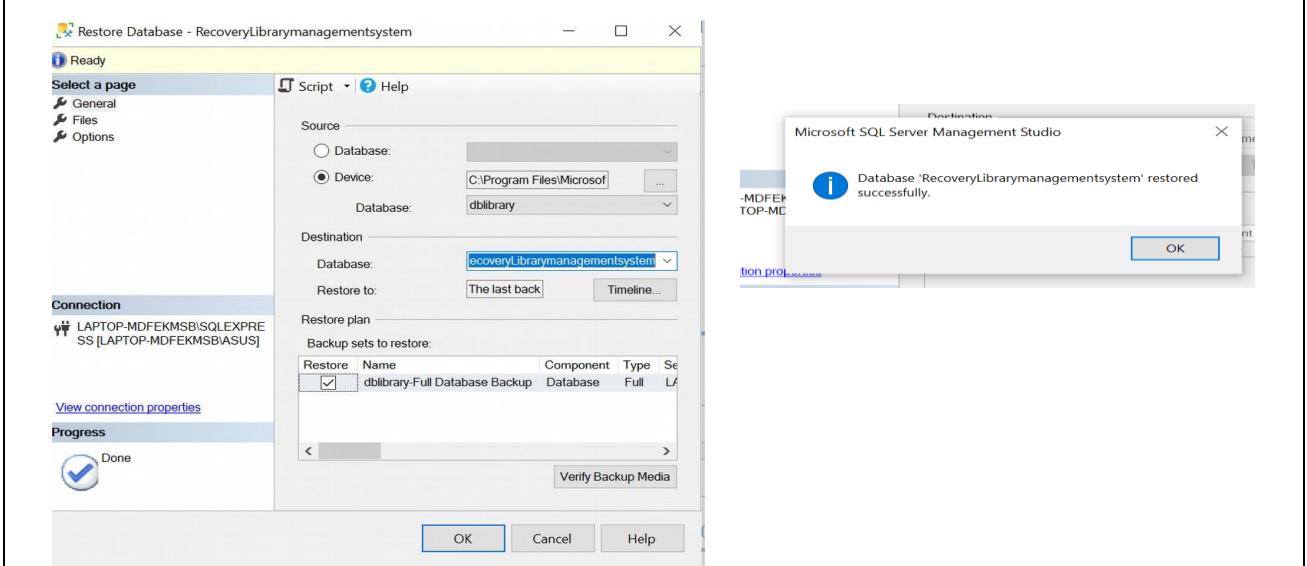
Step 1 Open Management studio and right click on database and press restore database.



Step 2: choose device option and add you .bak file and press on ok button



Step 3: given special name and press the ok button after than you have been successful to restore your database



Therefore, we can enable the system authorization for system security and also can add a role for users. I have shown how we can enable the system performance in the given document. In this part, I have also shown the database backup and restore to system security. With the help of system backup and restore, we can easily maintain our data safely.

### System administration:

The role of a system administration can be very diverse and the list of responsibilities can range from 5 hours a week to full time depending on the circumstances of the practice. Roles of system administration in LMS:

- System admin can also add, delete and update the book details LMS.
- System admin also can add, edit and delete the user details in LMS.
- System admin also can calculate the fine upon late return of book.

- System admin also accept, reject the book requested by the user.
- Ongoing monitoring and maintenance of both software and hardware systems.
- Ensuring up to date security measures are in use throughout your practice.
- Management of the network active directory account

A good systems administrator keeps your systems working smoothly on a day-to-day basis, but should also try to improve the efficiency of staff's use of technology and the systems in place throughout your practice. Some major benefits to having a systems administrator include:

- Confidence that your systems are stable, allowing more work to be achieved by staff.
- Lower maintenance and recovery costs.
- Able to identify and resolve problems before they cripple your business.
- Quick response times when something goes wrong.

### **System management**

Systems management is a broad umbrella term for several functions contributing to the way a system runs. Some organizations include network management within their definition of systems management, while others use the term to refer to the whole organization or department .Alternatively, some administrators might only use the term in reference to the management of data center.

These are some of the key functions of a systems management approach:

- Distribution of software
- Software upgrades
- Management of user profiles
- Version control
- Recovery and backup
- Performance planning
- Logs
- Automation
- System administration

Here are some of the general benefits you can expect:

- Increase in member satisfaction

### **Effectiveness of Database Management Tools**

I used SQL Service Profiler, Audit Logs, Extended Events, and other database administration tools for developing, securing, and checking the database for faults and problems. Intentional or inadvertent risks database security tools have also been used to protect and secure databases or stored procedures from unwanted access. Transactions, triggers, stored procedures, scheduled jobs, and different database-monitoring tools are among the database management techniques utilized. Schedule backup and recovery, authentication, and authorization are some of the database security technologies that have been employed in database systems. Those database management solutions offer a variety of advantages, but they also have their drawbacks.

## **Advantages of database management tools**

The database management tools provide us the various advantages like system performance optimization and to increase the security of database also. Below I have listed some of those benefits of database management tools:

The advantages of different management tools that have been used for the development of Library management system are listed below:

### **Stored Procedures**

- Stored procedures are cached on the server.
- Can be tested independent of the application.
- Scripts are in one location that makes updates and tracking of dependencies based on schema changes easier.
- Execution plans for the process are easily reviewable without having to run the application

### **Transactions**

- Data transaction help to maintain data integrity.
- Data reverse and change is quite easier.
- Useful for performing multiple operation against one or more table.

### **Triggers**

- Triggers can be used as an alternative method to check the integrity of data.
- SQL triggers provide an alternative way to run scheduled tasks.
- They are useful to audit the changes of data in tables.
- SQL triggers provide an alternative way to run scheduled tasks. You don't have to wait to run the scheduled tasks because the triggers are invoked.
- SQL triggers are useful to audit the changes of data in tables.

### **Transactions**

- The primary benefit of using transactions is data integrity. Many database uses require storing data to multiple tables or multiple rows to the same table in order to maintain a consistent data set.
- Using transactions ensures that other connections to the same database see either all the updates or none of them.
- Data reverse and change is quite easier.
- It is very useful while performing multiple operation against one or more table.

## SQL Profiler

- It can reveal how an instance works when it's interacting with a client.
- It lets users save trace data and compare it to newer data to spotlight new trouble spots.
- It can help zero in on trouble spots by allowing us to capture and replay key events
- This function also helps with stress testing and identifying slowly executing queries.
- Monitors the system performance in a better way.
- Provide advice on indexes that will boost performance.

## Authentication & Authorization

- It prevents the unauthorized access out of database as user will require username and password to access the database for modification.
- User accounts and all authentication are controlled by the database. There is no reliance on anything outside of the database.
- Microsoft SQL provides strong password management features to enhance security when using database authentication.

## Database Backup & Recovery

- The most significant benefit of data backup and recovery services is increased reliability.
- It easier to perform but is very effective method if somehow we lost our data.
- If we enable the schedule backup then we don't have to perform this task manually everyday.

## Shortcomings of Database Tools

### 1) Stored Procedures:

- i) Complex Stored Procedures that utilize complex, core functionality of the RDBMs used for their creation will not always port to upgraded versions of the same database.
- ii) Any data errors in handling Stored Procedures are not generated until runtime.
- iii) Depending on separation of concern there is the potential that Stored Procedure development could potentially require a dedicated database developer.

### 2) Transactions:

- i) If we design the transactions weakly then it can big negative impact on the scalability.
- ii) The use of transaction database engine can consume lots of space to store cache data that results in the slower performance of system.

### 3) Triggers:

- i) It is hard to follow their logic as they can be fired before or after the database insert/update happens.

- ii) It is easy to forget about triggers and if there is no documentation it will be difficult to figure out for new developers for their existence.
- iii) Triggers execute invisible to client-application application. They are not visible or can be traced in debugging code.

#### **4) SQL Profiler:**

- i) SQL Server Profiler is a GUI that utilizes SQL Server Trace through the client-side. Due to this nature, we can potentially see a mild to severe performance impact depending on the environment.
- ii) SQL Server Profiler is suited mainly for a quick glimpse at what is happening on the server. It is not intended to be run for long periods of time

#### **5) Database Backup & Recovery**

- i) It requires training to perform database backup and recovery. The individual with basic knowledge of database server can't perform it for the first time.
- ii) Sometimes our storage where we have the backups can become corrupt, and be rendered unusable.
- iii) Database administrators needs to do regular maintenance and checking in order for them to work properly.

#### **6) Authentication & Authorization**

- i) If the person who have created authentication forgets password or anyone misuses this feature in database then it may result to data loss.
- ii) Individual needs to provide the strong password as poor authentication and authorization slows the server response time.

While creating the Library management system various aspects like database maintenance and backup, application security and optimization have been implemented which will ensure the system effectiveness. As for now the system is small and handles only the small part of the college management but in future with the further use this system won't be able to handle all the queries and college may need to update this system. For now the system mostly performs the CRUD operations and provides only few logic in the application as the web application performing feature are all added to it. With the continuous and if college management wants future enhancement in the system then we can also do it as there are still various areas in this system for updates and improvement. This system now can create and store the information of college's books, user, available book and their number of copies etc. This system is very effective than the previous file system of the library management where library coordinator had to enter all the information about new available book release. In the above

documentation I have thoroughly analyzed and evaluated the effectiveness and proper report is generated with evidence which shows that all the client requirements are fulfilled.

The college management requirements have been met and the application performs smoothly but there are still some areas which we can further enhance later. There are many features and functions which can be added in this application to increase its security and performance. The application required for library management is very simple for now with only few logic and as it is a web application student still have to be physically in college to borrow or return the book. The security required for this application are pretty basic and I have also implemented the simple login for the system. Below I have discussed some of the improvement areas which I think can be added in further updates of application to increase its performance:-

- **Availability:** The execution of this software depends on the requirements so the system should meet the computer requirements. The application needs to be user friendly and technical and user documentation should be provided to the college management so that no mistakes or problems arise while handling the application. We have implemented the various database management tools and we need to regularly monitor and check their activities to increase the performance in application performance. As this system helps the library department store the information about books the system shouldn't be down while inserting or retrieving the information from database. For now we are using the community version of the SQL Server but in future to increase performance and get most of the feature from SQL Server we can also use the enterprise version which provides us more features fast than community version.
- **Scalability:** For now the library management system is only handled by the library department of college. However after sometime if the college administration to digitalize the college management then college has to deal with two applications. Instead of this the college can merge the library department with the overall college management so that all the features and functions of the college can be performed from single application. If it is made so then there will be required strong security and proper authorization also as there is various level of authorization in college and stores big data. Another alternative for the college administration in such case can be provided which is to create the web application for college that can handle all the functions of college online. While storing the information and functioning different department tasks the web application can also provide the information about college facilities and courses to online audience.
- **Performance:** The developed application performance is good, both the request and response time from server are average just like the web application's should be. As I have followed the

proper coding standards and implemented performance optimization tools like stored procedures, transactions and triggers in database the system is fast and performs normally. If I insert some complex queries to retrieve the information from database then it might increase the response time from server. So, we still need to look for further improvement in the performance of Library management application.

- **Security:** For the security of the application login form is created to protect and prevent the system access from malicious program and other threats. The admin is provided with all the access to system and users. If any individual try to gain unauthorized access to system then error message will be shown with proper validation. In database server also authentication is implemented and extended events like tools helps to properly monitor the activities of database. modification which helps to increase security in database. In future we can create more view in database to show to user show that users can't gain access to sensitive data of application. Another feature which we can implement to enhance the security of system is to implement encryption policy which can be adopted to protect the data.

### **Further Enhancements on System**

- Below I have mentioned some of the enhancements area in system, which can be implemented in future for the We can merge the library management system with the overall college administration system which will help in maintenance of the system and this way one application can handle all the functions of college.
- To improve the security of application various other types of authentication and authorization canbe implemented.
- Various other services and fields can be added in the application to increase the range of the application.
- Response time of the application must be increased.
- Scheduled backups can be used to safeguard the data for future use.
- Jobs logs and audit logs can be added in the maintenance section.
- Encryption, authentication and user manager policy can be used to improve security mechanism.
- User and technical documentation can be improved to help the users get compatible with the system.

## Conclusion

The implementation of different features presents in SQL server platform helped to handle concurrency, maintain security, and provided user authorizations and data recovery. The data concurrency in the system was handled implementing transaction. Similarly, various scheduled jobs and database backups were performed in order to prevent the catastrophic damage and accidents. This helped to recover the data in case of loss and attacks. The authorization and authentication were configured providing the specific roles to different users. Various database maintenance tools such as SQL error log, extended events, SQL server profiler etc. were implemented. Stored procedures triggers, audit log etc. were implemented effectively analyzing their shortcomings as well that leads to the development of effective system. Though the system had successfully addressed all the requirements set by the college, various enhancements can be made in the system in order to enhance the system features drastically.

## References

- Wikipedia. (2011). Retrieved from [https://en.wikipedia.org/wiki/Bellman%E2%80%93Ford\\_algorithm](https://en.wikipedia.org/wiki/Bellman%E2%80%93Ford_algorithm)
- Accounting Dictionary . (2018). *Accounting Dictionary* . Retrieved from www.myaccountingcourse.com: <https://www.myaccountingcourse.com/accounting-dictionary/business>
- BeginnersBooks. (2012). *BeginnersBooks*. Retrieved from <https://beginnersbook.com/2013/03/inheritance-in-java/>
- Dipak, P., Him, K., Kumar, K. M., Ramesh, R., & Kumar, S. R. (2012). *Computer Science* (4th ed.). Ananmagar: Buddha.
- fasproc. (2020). *fasproc*. Retrieved from fasproc.com: <https://fasproc.com/blog/business-process/>
- geeksforgeeks. (2020). Retrieved from geeksforgeeks: <https://www.geeksforgeeks.org/errors-v-s-exceptions-in-java/>
- GeeksforGeeks. (2020, May). Retrieved from [www.geeksforgeeks.org:](https://www.geeksforgeeks.org/data-structures/) <https://www.geeksforgeeks.org/data-structures/>
- hackerearth. (n.d.). *Algorithms*. Retrieved from hackerearth: <https://www.hackerearth.com/practice/algorithms/graphs/shortest-path-algorithms/tutorial/>
- Haskell. (2014, Dec). Retrieved from wiki.haskell.org: <https://wiki.haskell.org/Haskell>
- Javadoc. (2018). Retrieved from stack overflow: <https://stackoverflow.com/questions/912334/differences-between-exception-and-error/29898137>
- Margaret, R. (2011). *Techtarget*. Retrieved January 2011, from <https://searchsecurity.techtarget.com/definition/content-filtering>
- Margaret, R. (2018). *TechTarget*. Retrieved Feburary 2018, from <https://searchnetworking.techtarget.com/definition/bandwidth>
- NT, B. (2016). *5 advantages and disadvantages of Cloud Storage*. Retrieved April 1, 2018, from <http://bigdata-madesimple.com/5-advantages-and-disadvantages-of-cloud-storage/>

- Ram, S. L., & Kamal, B. (2005). *Computer Science* (3rd ed.). Bagbazar: Asia.
- skfdjskdfjs. (n.d.). *sdfsdf*. Retrieved 2014
- studytonight. (2020). Retrieved from studytonight: <https://www.studytonight.com/data-structures/queue-data-structure>
- techopedia. (2020). *Home/Dictionary /Tags/ Development* . Retrieved from techopedia: <https://www.techopedia.com/definition/21578/quicksort>
- techtarget. (2020). *techtarget*. Retrieved from searchbusinessanalytics.techtarget.com: <https://searchbusinessanalytics.techtarget.com/definition/business-intelligence-BI>
- Techtarget. (n.d.). *RODC (read-only domain controller)*. Retrieved from <https://searchwindowsserver.techtarget.com/definition/RODC-read-only-domain-controller>
- Tony, N., & J.V, M. (2011). *Microsoft* (2nd ed.). New Delhi: PHI learning Private.
- tutorialspoint. (2020). Retrieved from tutorialspoint: [https://www.tutorialspoint.com/java/java\\_exceptions.htm](https://www.tutorialspoint.com/java/java_exceptions.htm)
- tutorialspoint. (n.d.). *Data Structure - Bubble Sort Algorithm*. Retrieved from tutorialspoint: [https://www.tutorialspoint.com/data\\_structures\\_algorithms/bubble\\_sort\\_algorithm.htm](https://www.tutorialspoint.com/data_structures_algorithms/bubble_sort_algorithm.htm)
- w3schools. (n.d.). *Sorting Techniques*. Retrieved from w3schools: <https://www.w3schools.in/data-structures-tutorial/sorting-techniques/>
- Wilson, M. (2015). *Advantages and Disadvantages of Server Clustering*. Retrieved April 1, 2018, from <https://www.eukhost.com/blog/webhosting/advantages-and-disadvantages-of-server-clustering/>
- xgxdfd. (n.d.). *xxfdxd*. Retrieved 2010
- Yong. (2012). *NETWORK STANDARD*. Retrieved april 16, 2012, from <https://www.slideshare.net/makyong1/chapter-33-12564245>
- Zach, T. (2011). *Anandtech*. Retrieved September 2011, from <https://www.anandtech.com/show/4666/file-server-builders-guide>