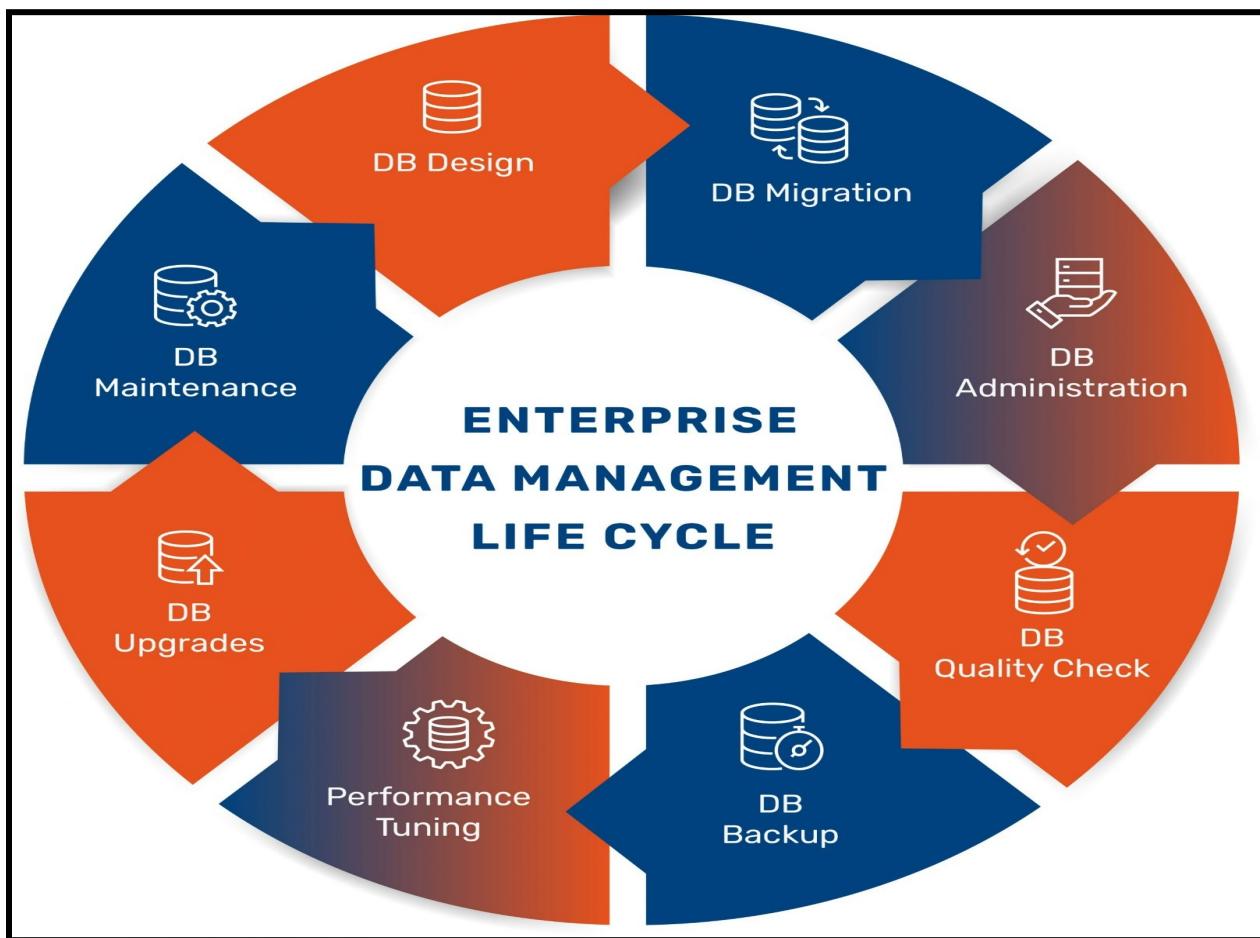


Online Data Storage Management System for Easy Reference



-by Group 4
Ayush Sharma,
Benji Babukutty Thomas,
Duy Thanh Nguyen,
Prakhar Gupta,
Pravali Chandragiri
Soumee Roy

Online Data Storage Management System for Easy Reference	0
1. Problem Statement	2
1.1 Problems without an Online data management storage system:	2
1.2 Competitor analysis	3
1.2.1 Drawbacks of competitors:	3
2. Proposed Solution	3
2.1 Business Scenario	4
2.2 Product Features:	4
2.3 Limitations for the project:	5
4. Assumptions	5
5. Relational Data Model	6
5.1 Tables	6
5.1.1 Accounting_T	6
5.1.2 Book_T	7
5.1.3 Chamber_Details_T	7
5.1.4 Emp_T	8
5.1.5 Groupm_T	8
5.1.6 Inventory_T	9
5.1.7 Publications_T	9
5.1.8 Writer_T	10
5.1.9 Transactions_T	11
5.1.10 Core_T	11
5.2 Audit Tables	12
5.2.1 Requests_T	12
5.2.2 BookToOrder_T	12
5.2.3 employee_audit	12
5.2.4 Groupm_audit	13
6. ER Diagram	14
7. Schema	15
8. Use cases	15
9. Stored Procedures	16
9.1: delete_Groupm	16
9.2 delete_employee	17
9.3: delete_books	17
9.4 add_books	19
9.5: Core_Entry_Borrowed_books	20

10. Triggers	22
10.1: employee_audit	22
10. 2: group_audit	23
10.3. insert_into_bookto_order	24
11. Functions	25
11.1: GetAvailableRooms()	25
11.2: GetMemberBooks()	25
11.3: GetPublisherBooks()	26
12. View	26
13. Sequence Diagram	27
13. General Queries:	28
14. What the Future Upholds?	28
15. Conclusion	29

1. Problem Statement

We are surrounded by data everywhere since it is said to be the new oil. In today's digital age, where the amount of data produced and stored is rapidly increasing, an online data storage management system is essential for individuals and organizations to manage their data effectively, efficiently, and securely.

1.1 Problems without an Online data management storage system:

Without proper data storage and management system, all big or small firms or individuals would feel lost and would face at least one of the below challenges:

- Inadequate storage space: Because of the storage space's limitations, vital information and documents could be lost or older files could need to be removed.
- Accessibility issues: If files are stored on different devices or are not centrally located, people may have trouble getting to them.
- Security issues: If people keep important data on personal devices or external hard drives, there is a possibility of security breaches, theft, or damage.
- Restrictions on collaboration: Teams may find it difficult to work together on projects or documents in the absence of a centralized storage system.
- Limited mobility: Without access to online storage, people could only access their files from a limited number of locations or devices.

1.2 Competitor analysis

Some of the established players in the domain are:

- Microsoft OneDrive for Business
- Google Drive Enterprise
- Dropbox Business
- Box Enterprise
- Amazon S3
- iCloud for Business
- pCloud Business
- Mega for Business
- Sync.com for Business
- DarwinBox

The same objective is shared by all of these online data management and storage solutions: to make it simple and accessible for users to access data when needed.

1.2.1 Drawbacks of competitors:

Even though they are established in the market, each product or service has its pros and cons. Some of the limitations that the above data management systems are facing and that our product would try to overcome are:

1. Limited sharing and collaboration features
2. Limited syncing
3. No client-side encryption by default
4. Limited platform support
5. Difficult to set up and use for non-technical users
6. Privacy concerns
7. Limited customizations

2. Proposed Solution

Data Sphere is a software development company that specializes in creating Database Management System (DBMS). We provide customized solutions to businesses of all sizes, from

small startups to large corporations. Our DBMS offers powerful and reliable data storage and retrieval capabilities, allowing businesses to manage their data with ease. We also provide a range of services, including data migration, backup and recovery, and performance tuning using Cloud services. With our expertise in DBMS development, we aim to help businesses improve their operations and drive growth. We primarily focus on enterprise or large-scale institutions.

2.1 Business Scenario

Our enterprise (Data Sphere) was founded in response to the growing demand for reliable and efficient DBMS solutions. As businesses generate more data than ever before, managing that data has become increasingly complex. We recognized the need for a more comprehensive solution that would enable businesses to store, retrieve, and analyze their data with ease.

2.2 Product Features:

Our DBMS is designed to be a powerful tool for businesses, with a range of features that make it a comprehensive solution for managing data. Some of the key features of our DBMS include:

Scalability: Our DBMS is designed to handle large volumes of data, making it a scalable solution for businesses of all sizes.

Security: We provide advanced security features, such as encryption and access controls, to protect our clients' data from unauthorized access or theft.



Data Types: Our DBMS supports a variety of data types, including structured and unstructured data, making it a versatile solution for businesses across industries.

Query Options: We provide a range of query options that enable users to extract insights from their data quickly, including SQL and NoSQL options.

Reports and Visualizations: Our system provides a range of reports and visualizations that help businesses identify trends and patterns in their data, allowing them to make data-driven decisions.

Integration: We offer integration with other software tools, such as business intelligence software, to provide a more comprehensive solution for our clients.

UI Implementation: While our DBMS does not require a UI to function, we recognize that a user-friendly interface can improve the user experience and make it easier for businesses to manage their data. We would recommend implementing a UI that provides users with an intuitive interface for managing their data. This could include features such as drag-and-drop data importing, customizable dashboards, and visualizations. We would also recommend implementing an alert system that notifies users of any potential issues with their data, such as anomalies or data quality issues.

Our enterprise has a team of experienced developers and data analysts who work closely with clients to understand their unique needs and develop customized solutions. We offer a range of services, including data migration, backup and recovery, and performance tuning, to ensure that our client's data is always available and secure.

2.3 Limitations for the project:

- UI and Cloud integration

3. Scenario

Here, for this project we have assumed that The University of Texas at Dallas has approached our firm (Data Sphere) for an efficient: Library Management storage system.

4. Assumptions

1. Books: Each book has a unique Book id number and a title. The same title may have multiple copies, but each copy has a unique identifier.
2. Members: Each member has a unique member identification number, and their name and address are stored in the system.
3. Publishers: Each publisher has a unique publisher identification number, and their names are stored in the system.
4. Employee: Each member has a unique employee identification number, and their name and address are stored in the system.

5. Chambers: Each room has a unique chamber identification number, and their names and availability are stored in the system.
6. Writer: Each author has a unique writer identification number, and their names are stored in the system.
7. Inventory: Each genre has a unique inventory identification number, and their names and quantities are stored in the system.
8. Relationships: There is a many-to-many relationship between books and authors, as a book may have multiple authors, and an author may have written multiple books. Publishers can publish multiple books, and a book can be published by multiple publishers. Borrowers can borrow multiple books, and a book can be borrowed by multiple borrowers.

Other assumptions regarding the project in real time:

1. Enterprise users will have a Dependence on Internet Connection: We presume that the institution employing our service will have a reliable internet connection because online data storage management systems require a fast and reliable internet connection.
2. Data Privacy: The institution employing the online data storage management systems would be willing to share their data in accordance with data protection laws and would not raise concerns about data privacy and security.
3. Potential Downtime: There wouldn't be any server outage or malfunction that would cause the online data storage management systems to go down and make it impossible for users to access their data.
4. Limited Controls and customizations: Despite the fact that we offer controls and customizations, we believe that the needs of organizations using our online data storage management systems will be met by these.

5. Relational Data Model

5.1 Tables

We have 10 tables in our online data storage management system for UTD's library management system:

5.1.1 Accounting_T

- This table is responsible for storing the details of the types of accounts available, from where we record the finances.
- Here we have considered Acc_ID i.e., Account Id as Primary Key.

	Acc_ID	Account_Type	Tariffindoll...	
►	1	Overdue	50	
	2	Non attendance	30	
	3	Vandalised	40	
	4	Lost	80	
	5	New Member Fee	30	
	6	Membership renewal	25	
	7	Existing Member	0	
	8	Late fee	20	
	9	Delivery fee	10	
	10	Processing fee	15	
	11	Reservation fee	5	
	12	Return fee	15	

5.1.2 Book_T

- This table is responsible for storing the details of the books available at the UTD library.
- Here we have considered B_id i.e., Book Id as Primary Key.

B_id	B_Name	Writer_id	Pub_id	Inv_id	is_available
► 1	The Shining	1	2	5	0
2	Pet Sematary	1	6	15	0
3	The Alchemist	2	3	4	0
4	Veronika Decides to Die	36	36	14	0
5	The Zahir	2	5	24	1
6	By the River Piedra I Sat Down and Wept	37	50	34	1
7	The Poems of Emily Dickinson	3	40	44	0
8	Harry Potter	4	45	1	0
9	Murder on the Orient Express	5	1	16	0
10	Death on the Nile	5	19	6	1
11	The Adventures of Huckleberry Finn	6	25	1	0
12	The Adventures of Tom Sawyer	8	25	44	0

5.1.3 Chamber_Details_T

- This table is responsible for storing the rooms or offices available in the organization i.e., UTD Library.
- Here ChamberID i.e., the id of the chamber or chamber number as Primary Key.

ChamberID	ChamberType	Chamber_Cap	Chamber_availability
1	Alamo, MC 3.442-Study Rooms	4	0
2	Bluebonnet, MC 3.618-Study Rooms	6	1
3	Honeybee, MC 3.710-Study Rooms	8	1
4	Horned Lizard, MC 3.228-Study Rooms	4	1
5	Mindfulness Room (MC 2.518)-Mindfulness Room	2	0
6	Smart Armadillo Room (MC 2.512)-Technology Space	10	0
7	Smart Chili Room (MC 3.701)-Technology Spaces	4	1
8	Smart Tejas Room (MC 3.704)-Technology Space	10	1
9	Smart Friendship Room (MC 2.530)-Technology Space	14	0
10	Prickly Pear, MC 2.520-Study Rooms	5	0
11	Topaz, MC 3.236-Study Rooms	4	1
12	Rodeo, MC 3.230-Study Rooms	5	0

5.1.4 Emp_T

- This table is responsible for storing the employee details, who are working at the UTD library.
- Here, Emp_id i.e., the ID assigned to each employee, is considered the primary key.

Emp_id	Emp_FirstName	Emp_LastName	Emp_Address	Emp_City	Emp_St	Emp_Zip	Emp_Num	Emp_isAvailable
1	Prakhar	Gupta	280 W Renner Rd	Richardson	TX	75080	94526362636	1
2	Kislay	Trivedi	New Hampshire	Chicago	IL	67587	94526362636	0
3	Abhay	Singh	250 Preston Rd	Richardson	TX	75080	2464571234	0
4	Divij	Karwasara	980 Horsehoe Drive	New York City	NY	82467	7562471358	1
5	Puneet	Sangwan	25th Ave Road	Seattle	WA	98121	9452636167	1
6	Christopher	Lee	864 Cherry Lane	Seattle	WA	98794	9873216549	1
7	Samantha	Clark	975 Oakwood Boulevard	Seattle	WA	98165	9456123785	0
8	Allen	Garcia	7650 McCallum Blvd	Dallas	TX	75007	9564543214	1
9	Jayt	Brown	1830 Post OAK Lane	Carrolton	TX	78447	9856231475	1
10	Earnest	Drone	12400 Montana Dr	Frisco	TX	97458	99154623784	0
11	Aeiron	Washington	9301 Fairmont Dr	Denton	TX	67845	9685741239	1
12	Iohn	Dos	8400 Blackthorn Trl	Houston	TX	654122	0764215597	0

5.1.5 Groupm_T

- This table is responsible for storing the details of the members registered at the UTD library.
- Here, GroupmID, i.e., the ID assigned to each member, is considered the primary key.

GroupmID	GroupmName	Groupm_EmailId	Groupm_Num	Groupm_PREFERREDCONTACTMETHOD	is_active
1	Ayush Sharma	ayush.sharma@gmail.com	9564563214	Mobile	1
2	Prakhar Gupta	prakhar.gupta@gmail.com	8529669258	Email	1
3	Soumy Roy	soumee.roy@gmail.com	7412552147	Mobile	1
4	Benji Thomas	benji.thomas@gmail.com	9632552369	Email	1
5	Pravali Chandragiri	pravali.chandragiri@gmail.com	7206958110	Mobile	1
6	Duy Nguyen	duy.nguyen@gmail.com	7894554987	Email	1
NULL	NULL	NULL	NULL	NULL	NULL

5.1.6 Inventory_T

- This table is responsible for storing the genre details available at the UTD Library.
- Here, Item_ID, i.e., the ID assigned to each genre, is considered the primary key.

Item_ID	Item_Category	Item_Quantity	Latest_ItemDate_Receiv...
1	Adventure	67	2023-05-01 05:17:24
2	Romance	300	2023-05-01 05:17:24
3	History	23	2023-05-01 05:17:24
4	Fictional	567	2023-05-01 05:17:24
5	Horror	768	2023-05-01 05:17:24
6	Mystery	257	2023-05-01 05:17:24
7	Non-fictional	299	2023-05-01 05:17:24
8	Tragedy	523	2023-05-01 05:17:24
9	Fantasy	876	2023-05-01 05:17:24
10	Science Fiction	456	2023-05-01 05:17:24
11	Memoir	234	2023-05-01 05:17:24
12	Biography	342	2023-05-01 05:17:24

5.1.7 Publications_T

- This table is responsible for storing the publications names whose books UTD owns.
- Here, PublicationID i.e, the ID assigned to each publication is considered the primary key.

	PublicationID	PublicationName
►	1	Penguin
►	2	Pandas
►	3	WritingCompany
►	4	GoodFolks
►	5	StoryTeller
►	6	BlueBird
►	7	HarperCollins
►	8	Simon & Schuster
►	9	Hachette Book Group
►	10	Random House
►	11	Macmillan Publishers
►	12	Scholastic Corporation

5.1.8 Writer_T

- This table is responsible for storing the authors' names whose books UTD owns.
- Here, Writer_ID i.e, the ID assigned to each writer, is considered the primary key.

	Writer_ID	WriterName
►	1	Stephen King
►	2	Paulo Coelho
►	3	Emily Dickinson
►	4	J.K. Rowling
►	5	Agatha Christie
►	6	Mark Twain
►	7	Anne Rice
►	8	Jane Austen
►	9	William Shakespeare
►	10	Leo Tolstoy
►	11	Emily Temple
►	12	George Orwell

5.1.9 Transactions T

- This table is responsible for storing all the transactions made at the UTD library.
 - Here, Transaction_ID i.e, the ID assigned to each Transaction made is considered the primary key.

	Transaction_ID	Transaction_Type	Fees_Type	Groupm_ID
▶	1	Checkout	Late fee	1
	2	Check-in	Processing fee	2
	3	Renewal	Lost item fee	3
	4	Reservation	Late fee	4
	5	Checkout	Processing fee	5
	6	Check-in	Lost item fee	6
	7	Renewal	Late fee	7
	8	Reservation	Processing fee	8
	9	Checkout	Late fee	9
	10	Check-in	Lost item fee	10
	11	Renewal	Processing fee	11
	12	Reservation	Late fee	12

5.1.10 Core_T

- This table is considered as the master table for the database.
 - Here, TransactionID i.e, the ID assigned to each request made is considered the primary key..

5.2 Audit Tables

We have 4 audit tables to track changes of data in the database:

5.2.1 Requests_T

- This table is responsible for storing the books which have been checked out or borrowed from the UTD library.
- Here, RequestID i.e, the ID assigned to each request made is considered the primary key.

	RequestID	GroupmID	Book_Name
▶	1	1	The Shining
◀	2	2	Pet Semetary
◀	3	3	The Alchemist
◀	4	4	Veronika Decides to Die
◀	5	5	The Zahir
	NULL	NULL	NULL

5.2.2 BookToOrder_T

- The books that UTD Library will need to order are listed in this table.
- This table is updated when a borrower checks out a book more than five times, to indicate that the library should order that specific order.
- Here we have considered OrderID i.e., Account Id as Primary Key.

	OrderID	Book_Name
▶	NULL	NULL
◀		
◀		
◀		

5.2.3 employee_audit

- The modifications that the employees make to the database will be recorded in this table. All historical data along with date and time will be stored here.

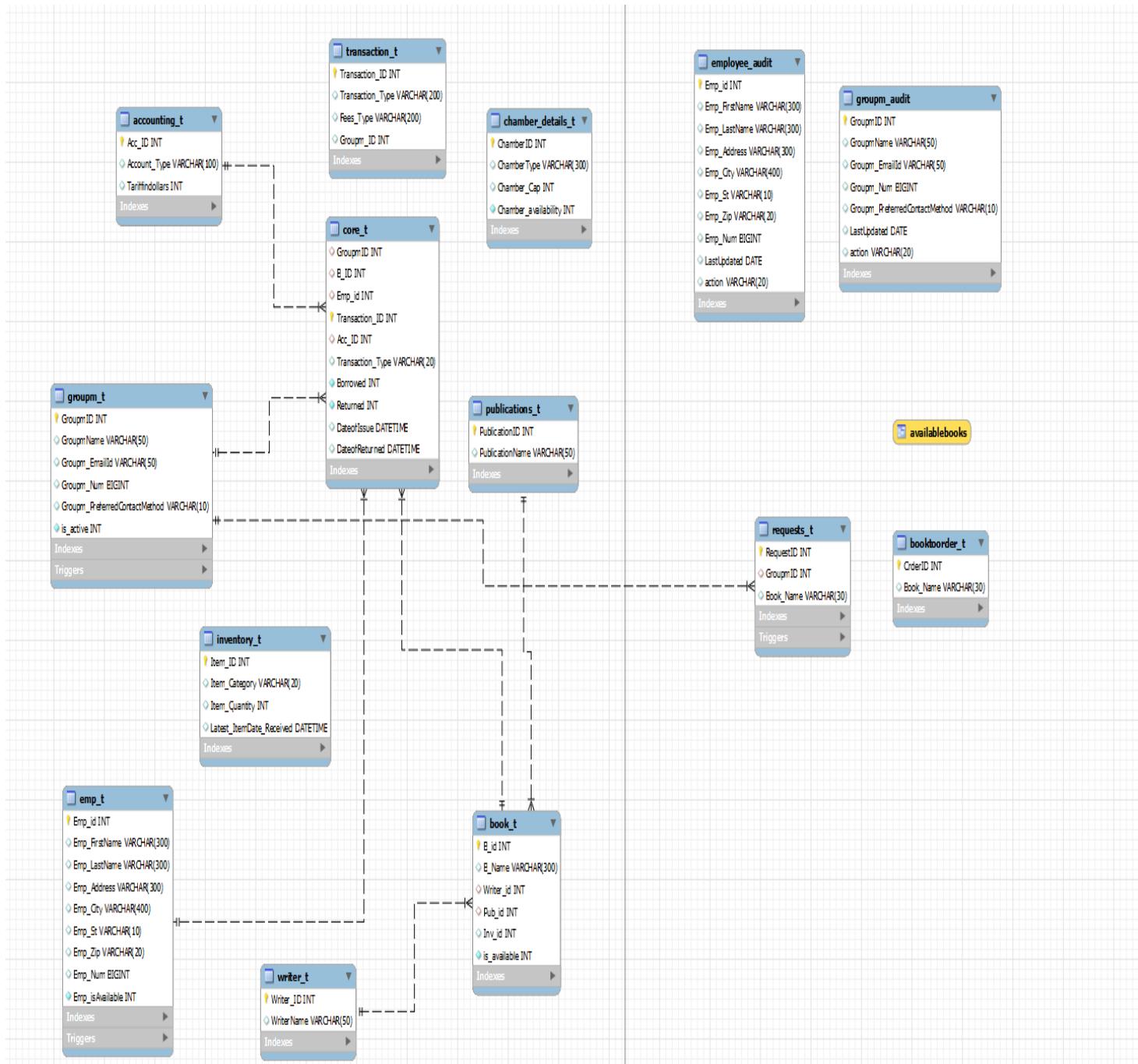
- Here, Emp_id i.e., the ID assigned to each employee, is considered the primary key.

5.2.4 Groupm_audit

- The modifications that the members of the library make to the database will be recorded in this table. All historical data along with date and time will be stored here.
 - Here, GroupmID, i.e., the ID assigned to each member, is considered the primary key.

GroupmID	GroupmName	Groupm_EmailId	Groupm_Num	Groupm_PREFERREDCONTACTMETHOD	LastUpdated	action
▶ NULL	NULL	NULL	NULL	NULL	NULL	NULL

6. ER Diagram



7. Schema

- N books may be reserved by a member, but only one reader may reserve any given book. Hence has 1:N relationship.
- Since a publisher can publish many books, but a book can only be published by one publisher, they have a 1:N relationship.
- The relation is 1:N, meaning that a group member may make N bookings for rooms, but only 1 person may reserve each room.
- An employee can help several members but only one employee will help each member in a 1:N connection.

8. Use cases

Use case 1: Borrowing a book and returning it.

When a customer borrows a book, the system inserts the data in the core table and checks for groupm id, staff id , and b_id. If all three details are valid it inserts the data into the core table. Else, displays an invalid detail message.

When a customer returns a book, the system updates in the core table and checks for the groupm id and book id in the core table. If the groupm id and book id exist in the core table then it removes the information of the book from the list of borrowed books and updates the list of returned books with the book's information. It also updates the employee details who was in charge of the service when the book was returned.

Use case 2: Adding a new book

When a new book is added, the system updates the book and the inventory table. It checks if the writer is a new writer or existing writer and if the writer is new, it adds the details in the writer table. Also, it checks if the publisher is new or existing and if the publisher is new, it adds the details to the publisher table. In this method, the library management system can sort out books with writer information and publisher information.

Use case 3: Updating Group member information

In order to enhance the efficiency of the library management system, we maintain a record of our group members' information in the Groupm table. If a member is deleted, the Groupm table indicates that the member is no longer active. Additionally, when we update member

information, we insert the old information into the Groupm audit table to ensure that their information is retained in case it is needed in the future.

9. Stored Procedures

9.1: delete_Groupm

We define a stored procedure called "delete_Groupm" that takes an input parameter called "Groupm_ID". The stored procedure updates the "is_active" column in the "Groupm_T" table to 0 for the row where the "Groupm_ID" column matches the input parameter value, marking the group as inactive or deleted. After a member is removed, the member table will undergo an update process where the is_active column will be modified. This column will display a value of 0 to indicate that the member is now inactive.

```
/*Delete group member from the list*/
CALL delete_Groupm(4);
CALL delete_Groupm(5);
SELECT * FROM Groupm_T;
```

To understand the query we took an example where we are deleting the 2 rows i.e., row 4 and row 5.

	GroupmID	GroupmName	Groupm_EmailId	Groupm_Num	Groupm_PREFERREDCONTACTMETHOD	is_active
▶	1	Ayush Sharma	ayush.sharma@gmail.com	9564563214	Mobile	1
	2	Prakhar Gupta	prakhar.gupta@gmail.com	8529669258	Email	1
	3	Soumee Roy	soumee.roy@gmail.com	7412552147	Mobile	1
	4	Benji Thomas	benji.thomas@gmail.com	9632552369	Email	0
	5	Pravali Chandragiri	pravali.chandragiri@gmail.com	7206958110	Mobile	0
	6	Duy Nguyen	duy.nguyen@gmail.com	7894554987	Email	1

9.2 delete_employee

This query creates a stored procedure called "delete_employee" which takes an input parameter "Employee_ID". The stored procedure updates the "Emp_isAvailable" column in the "emp_t" table to 0 for the row where the "Employee_ID" column matches the input parameter value. This means the employee with that ID is marked as unavailable or deleted in the system.

```

47 • CALL delete_employee(2);
48 • call delete_employee(3);
49 • SELECT * FROM emp_t;

```

By calling the procedure we are trying to delete the employee ID 2 and 3. After the procedure executes it sets the ID 2 and 3 availability is 0.

	Emp_id	Emp_FirstName	Emp_LastName	Emp_Address	Emp_City	Emp_St	Emp_Zip	Emp_Num	Emp_isAvailable
▶	1	Prakhar	Gupta	280 W Renner Rd	Richardson	TX	75080	94526362636	1
	2	Kislay	Trivedi	New Hampshire	Chicago	IL	67587	94526362636	0
	3	Abhay	Singh	250 Preston Rd	Richardson	TX	75080	2464571234	0
	4	Divij	Karwasara	980 Horsehoe Drive	New York City	NY	82467	7562471358	1
	5	Puneet	Sangwan	25th Ave Road	Seattle	WA	98121	9452636167	1

9.3: delete_books

We created a stored procedure "delete_books" with an input parameter "Book_ID". This procedure modifies the "is_available" column in the "book_t" table by setting it to 1 for the row where the "Book_ID" column matches the input parameter value, indicating that the book with that ID is available in the system.

Furthermore, the procedure updates the "Item_Quantity" column to 0 and sets the "Latest_ItemDate_Received" to the current date and time in the "Inventory_T" table for the row where the "Item_ID" column matches the input parameter value. This results in the removal of the book from the inventory.

```
CALL delete_books(2);
SELECT * FROM book_t;
```

To execute the "delete_books" stored procedure using a parameter value of 2 for "Book_ID", we used the "CALL" statement. This will update the "book_t" and "Inventory_T" tables such that the book with an ID of 2 is marked as available and removed from the inventory table.

before execution:

book_t:

	B_id	B_Name	Writer_id	Pub_id	Inv_id	is_available
▶	1	The Shining	1	2	5	0
	2	Pet Sematary	1	6	15	0
	3	The Alchemist	2	3	4	0
	4	Veronika Decides to Die	36	36	14	0
	5	The Zahir	2	5	24	0
	6	By the River Piedra I Sat Down and Wept	37	50	34	0
	7	The Poems of Emily Dickinson	3	40	44	0
	8	Harry Potter	4	45	1	0

Inventory_t:

	Item_ID	Item_Category	Item_Quantity	Latest_ItemDate_Received
▶	1	Adventure	66	2023-05-01 09:21:17
	2	Romance	299	2023-05-01 09:21:17
	3	History	22	2023-05-01 09:21:17
	4	Fictional	566	2023-05-01 09:21:17
	5	Horror	767	2023-05-01 09:21:17
	6	Mystery	256	2023-05-01 09:21:17
	7	Non-fictional	298	2023-05-01 09:21:17
	8	Tragedy	522	2023-05-01 09:21:17

after execution:

book_t:

	B_id	B_Name	Writer_id	Pub_id	Inv_id	is_available
	2	Pet Semetary	1	6	15	1
	3	The Alchemist	2	3	4	0
	4	Veronika Decides to Die	36	36	14	0
	5	The Zahir	2	5	24	0
	6	By the River Piedra I Sat Down and Wept	37	50	34	0

inventory_t:

	Item_ID	Item_Category	Item_Quantity	Latest_ItemDate_Received
▶	1	Adventure	66	2023-05-01 09:21:17
	2	Romance	0	2023-05-01 09:23:33
	3	History	22	2023-05-01 09:21:17
	4	Fictional	566	2023-05-01 09:21:17
	5	Horror	767	2023-05-01 09:21:17
	6	Mystery	256	2023-05-01 09:21:17
	7	Non-fictional	298	2023-05-01 09:21:17
	8	Traoedv	522	2023-05-01 09:21:17

9.4 add_books

This SQL code creates a stored procedure called "add_books". The procedure takes three input parameters and inserts the Publisher_name and Writer_Name into their respective tables if they do not exist. It then inserts the Book_Name along with the related Writer_ID and Pub_id into the "book_t" table. If Book_Name already exists in "book_t", it selects "Book already in database".

```
CALL add_books("Pet Sematary", "Stephen King", "Pandas");
SELECT * FROM book_t;
```

Here we are passing the 3 book names values into the the procedure to store it in the book table.

before execution:

	B_id	B_Name	Writer_id	Pub_id	Inv_id	is_available
▶	1	The Shining	1	2	5	0
	2	Pet Sematary	1	6	15	1
	3	The Alchemist	2	3	4	0
	4	Veronika Decides to Die	36	36	14	0
	5	The Zahir	2	5	24	0
	6	By the River Piedra I Sat Down and Wept	37	50	34	0
	7	The Poems of Emily Dickinson	3	40	44	0
	8	Harry Potter	4	45	1	0

after execution:

Result Grid		Filter Rows:
	Book already in database	
▶	Book already in database	

9.5: Core_Entry_Borrowed_books

This SQL code creates a stored procedure called "Core_Entry_Borrowed_books". The procedure checks whether the quantity of the item in the inventory corresponding to the given "Book_ID" is greater than or equal to one and if the "Book_ID" is present in the inventory.

If the condition is true, the procedure inserts a new row into the "core_t" table .Additionally, it updates the "Item_Quantity" and "Latest_ItemDate_Received" columns in the "Inventory_T" table by subtracting 1 row the "Item_Quantity" and setting the "Latest_ItemDate_Received" to NOW().

If the condition is false, the procedure selects "Book out of stock" message.:.

```
Call Core_Entry_Borrowed_books(1,1,1);
/*TRUNCATE TABLE core_t;*/
select * from core_t;
```

```
SELECT * FROM transaction_t;
```

By calling the above procedure it will execute and create a new row in the core table.

before execution:

core t:

transaction_t:

	Transaction_ID	Transaction_Type	Fees_Type	Groupm_ID
▶	1	Checkout	Late fee	1
	2	Check-in	Processing fee	2
	3	Renewal	Lost item fee	3
	4	Reservation	Late fee	4
	5	Checkout	Processing fee	5
	6	Check-in	Lost item fee	6
	7	Renewal	Late fee	7
	8	Reservation	Processing fee	8

After execution:

10. Triggers

10.1: employee_audit

This trigger will invoke once when we update the emp table.

```
UPDATE emp_t
SET
    Emp_FirstName = "Jared" ,
    Emp_LastName = "Smith"
WHERE
    Emp_id = 3;
```

Here we are updating the emp_t. So, when we update the emp_t, the above trigger will invoke and the previous records will be seen in the employee audit table

Before trigger:

Result Grid		Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:	
	Emp_id	Emp_FirstName	Emp_LastName	Emp_Address	Emp_City	Emp_St	Emp_Zip	Emp_Num	Emp_isAvailable
▶	1	Prakhar	Gupta	280 W Renner Rd	Richardson	TX	75080	94526362636	1
	2	Kislay	Trivedi	New Hampshire	Chicago	IL	67587	94526362636	0
	3	Abhay	Singh	250 Preston Rd	Richardson	TX	75080	2464571234	0
	4	Divij	Karwasara	980 Horsehoe Drive	New York City	NY	82467	7562471358	1
	5	Puneet	Sangwan	25th Ave Road	Seattle	WA	98121	9452636167	1
	6	Christopher	Lee	864 Cherry Lane	Seattle	WA	98794	9873216549	1
	7	Samantha	Clark	975 Oakwood Boulevard	Seattle	WA	98165	9456123785	0
	8	Allen	Garcia	7650 McCallum Blvd	Dallas	TX	75007	9564543214	1

after trigger:

	Emp_id	Emp_FirstName	Emp_LastName	Emp_Address	Emp_City	Emp_St	Emp_Zip	Emp_Num	Emp_isAvailable
▶	1	Prakhar	Gupta	280 W Renner Rd	Richardson	TX	75080	94526362636	1
	2	Kislay	Trivedi	New Hampshire	Chicago	IL	67587	94526362636	0
	3	Jared	Smith	250 Preston Rd	Richardson	TX	75080	2464571234	0
	4	Divij	Karwasara	980 Horseshoe Drive	New York City	NY	82467	7562471358	1
	5	Puneet	Sangwan	25th Ave Road	Seattle	WA	98121	9452636167	1
	6	Christopher	Lee	864 Cherry Lane	Seattle	WA	98794	9873216549	1
	7	Samantha	Clark	975 Oakwood Boulevard	Seattle	WA	98165	9456123785	0
	8	Allen	Garcia	7650 McCallum Blvd	Dallas	TX	75007	9564543214	1
	emp_t 4								

employee_audit:

	Emp_id	Emp_FirstName	Emp_LastName	Emp_Address	Emp_City	Emp_St	Emp_Zip	Emp_Num	LastUpdated	action
▶	3	Abhay	Singh	250 Preston Rd	Richardson	TX	75080	2464571234	2023-05-01	Updated
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

10. 2: group_audit

```

/*Using trigger for updating the table*/
UPDATE Groupm_T
SET
    GroupmName = "Soumee Roy"
WHERE
    GroupmID = 3;

SELECT * FROM Groupm_T;

SELECT * FROM Groupm_audit ;

```

When the group table is updated in the group members table, groupm_audit trigger will be invoked and member_audit table will be inserted with the old records of group table.

Before Trigger:

	GroupmID	GroupmName	Groupm_EmailId	Groupm_Num	Groupm_PreferredContactMethod	is_active
▶	1	Ayush Sharma	ayush.sharma@gmail.com	9564563214	Mobile	1
	2	Prakhar Gupta	prakhar.gupta@gmail.com	8529669258	Email	1
	3	Soumy Roy	soumee.roy@gmail.com	7412552147	Mobile	1
	4	Benji Thomas	benji.thomas@gmail.com	9632552369	Email	0
	5	Pravali Chandragiri	pravali.chandragiri@gmail.com	7206958110	Mobile	1

After trigger:

	GroupmID	GroupmName	Groupm_EmailId	Groupm_Num	Groupm_PreferredContactMethod	is_active
	1	Ayush Sharma	ayush.sharma@gmail.com	9564563214	Mobile	1
	2	Prakhar Gupta	prakhar.gupta@gmail.com	8529669258	Email	1
	3	Soumee Roy	soumee.roy@gmail.com	7412552147	Mobile	1
	4	Benji Thomas	benji.thomas@gmail.com	9632552369	Email	0
	5	Pravali Chandragiri	pravali.chandragiri@gmail.com	7206958110	Mobile	1

Member_audit

table:

	GroupmID	GroupmName	Groupm_EmailId	Groupm_Num	Groupm_PreferredContactMethod	LastUpdated	action
▶	3	Soumee Roy	soumee.roy@gmail.com	7412552147	Mobile	2023-05-01	Updated
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

10.3. insert_into_bookto_order

Here, the trigger will be invoked when the count of the group names in the book table is above 5.

before trigger:

book_to_order table

	OrderID	Book_Name
*	NULL	NULL

Once the record record is greater than 5, it inserts the new book details into the book to order table.

11. Functions

11.1: GetAvailableRooms()

Returns the number of available rooms in the chamber_details_t table where the Chamber_availability column is equal to 1.

One can see the availability of the chamber rooms.

	ChamberID	ChamberType	Chamber_Cap	Chamber_availability
▶	1	Alamo, MC 3.442-Study Rooms	4	0
	2	Bluebonnet, MC 3.618-Study Rooms	6	1
	3	Honeybee, MC 3.710-Study Rooms	8	1
	4	Horned Lizard, MC 3.228-Study Rooms	4	1
	5	Mindfulness Room (MC 2.518)-Mindfulness Roo...	2	0
	6	Smart Armadillo Room (MC 2.512)-Technology S...	10	0
	7	Smart Chili Room (MC 3.701)-Technology Spaces	4	1
	8	Smart Tejas Room (MC 3.704)-Technology Spaces	10	1
	9	Smart Friendship Room (MC 2.530)-Technology ...	14	0
	10	Prickly Pear, MC 2.520-Study Rooms	5	0
	11	Topaz, MC 3.236-Study Rooms	4	1
	12	Rodeo, MC 3.230-Study Rooms	5	0
*	NONE	NONE	NONE	NONE

```
/* Function*
/*To find the number of available rooms*/
SELECT * from chamber_details_t;

SELECT GetAvailableRooms() as number_of_rooms_Available;
```

	number_of_rooms_Available
▶	6

11.2: GetMemberBooks()

This function returns a string containing the names of all books borrowed by a member identified by their GroupmID in the core_t and book_t tables.

	groupmName	groupm_emailID	b_name
▶	Ayush Sharma	ayush.sharma@gmail.com	The Zahir
	Ayush Sharma	ayush.sharma@gmail.com	Pet Sematary
	Ayush Sharma	ayush.sharma@gmail.com	The Shining
	Prakhar Gupta	prakhar.gupta@gmail.com	NULL
	Soumy Roy	soumee.roy@gmail.com	NULL
	Benji Thomas	benji.thomas@gmail.com	NULL
	Pravali Chandragiri	pravali.chandragiri@gmail.com	NULL
	Duy Nguyen	duy.nguyen@gmail.com	NULL

```
/*
 *Get the books for a particular group member*
SELECT GetMemberBooks(1);
```

	GetMemberBooks(1)
▶	The Shining, Pet Sematary, The Zahir

11.3: GetPublisherBooks()

This function retrieves the names of books published by a specific publisher, based on the publisher_id passed as a parameter.

The book_t table has a foreign key referencing the publications_t table with a column named "Pub_id".

```
/*
 *Get books that are published by a particular publisher*
SELECT GetPublisherBooks(3);
```

	GetPublisherBooks(3)
▶	The Alchemist The Hitchhiker's Guid...

The Alchemist
The Hitchhiker's Guide to the Galaxy
The Hobbit
The Lord of the Rings

12. View

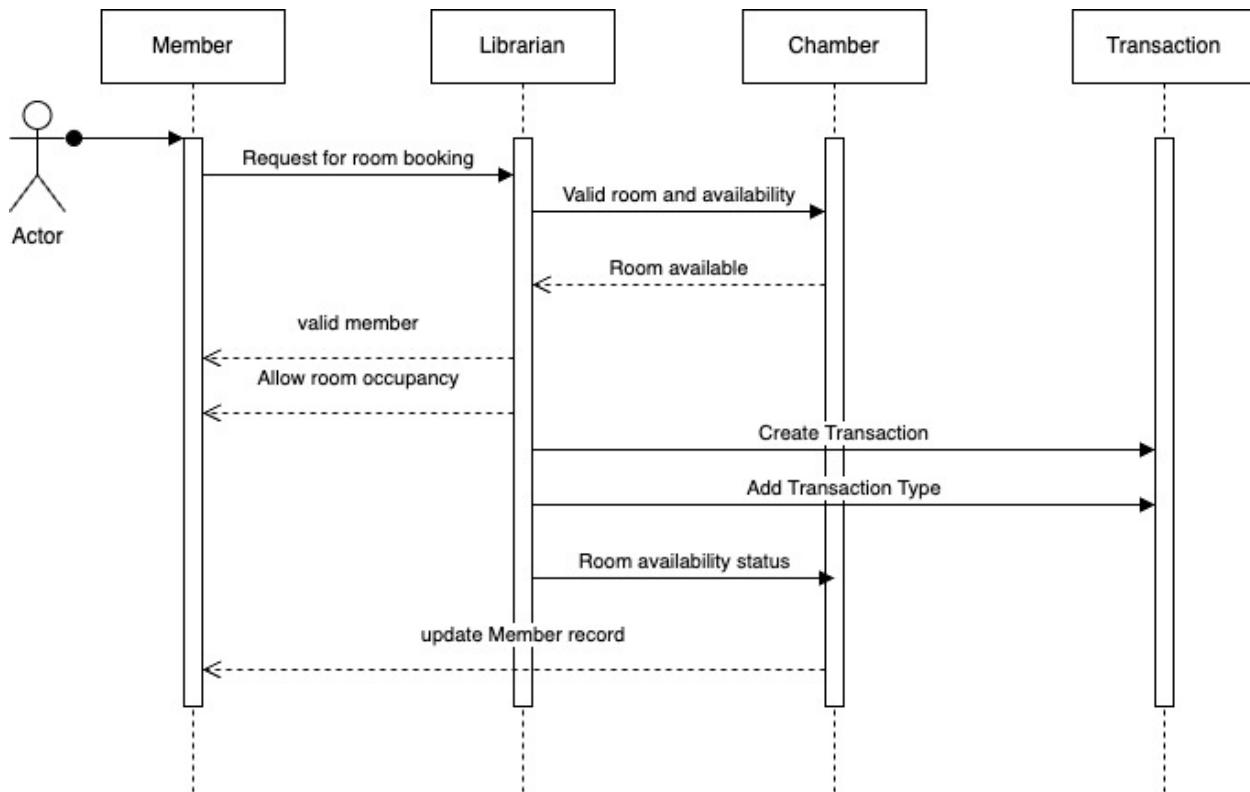
AvailableBooks:

A view that lists the details of all books that are currently available in the library, along with the name of the author who wrote them and the name of the publisher who published them.

```
/*View*/  
  
SELECT * FROM AvailableBooks;
```

B_id	book_name	Writer_name	publisher_name
5	The Zahir	Paulo Coelho	StoryTeller
6	By the River Piedra I Sat Down and Wept	Octavia Butler	Johns Hopkins University Press
10	Death on the Nile	Agatha Christie	Wiley
14	The Vampire Lestat	Anne Rice	University of Chicago Press
15	Pride and Prejudice	Jane Austen	New Directions Publishing
16	Sense and Sensibility	Isaac Asimov	Melville House
31	The Count of Monte Cristo	James Baldwin	Bloomsbury Publishing
34	The Lord of the Rings	Charles Dickens	WritingCompany
36	To Kill a Kingdom	J.D. Salinger	Little, Brown and Company
37	The Joy Luck Club	Kurt Vonnequt	Verso Books
38	The Color Purple	The Joy Luck Club	aymarket Books
39	The Immortal Life of Henrietta Lacks	Virginia Woolf	And Other Stories
41	The Adventures of Sherlock Holmes	Mary Shelley	Perseus Books Group
42	The Memoirs of Sherlock Holmes	Maya Angelou	Macmillan Publishers

13. Sequence Diagram



13. General Queries:

The query to select the book ID, book name, writer ID, and availability status of books in the book table that have corresponding entries in the core table.

The names and email IDs of the group members who have borrowed a book and not returned it yet, along with the book ID, transaction ID, and the expected return date.

Get the total number of books in the database that were published by "Penguin" and have a genre of "Romance"

14. What the Future Upholds?

Increased automation: In the future, online data storage management systems are probably going to get more automated. This can involve scheduling automatic backups, organizing files, or even labeling files with keywords for quick access.

Cloud-based storage: In the future, cloud-based storage is probably going to become even more common. This will increase scalability and cost-effectiveness for enterprises as well as give users more freedom to access data from any location.

Artificial intelligence: Online data storage and management systems are likely to use artificial intelligence (AI) more frequently. This might include tools like predictive analytics, customized suggestions for file organization, and even AI-powered assistants that can make it simple and quick for users to find and retrieve the files they require.

Data security is becoming increasingly important, and online data storage management systems will continue to give security tools like encryption and multi-factor authentication top priority. In addition, the security of online data storage systems will be improved using blockchain technology.

Integration with other systems: Systems for managing online data storage are likely to be more integrated with other systems, such as software for managing projects and communication tools. Businesses will be able to collaborate more effectively and streamline procedures as a result.

15. Conclusion

Our enterprise offers a powerful and comprehensive Online DBMS solution for businesses of all sizes. Our system is designed to be scalable, secure, and versatile, with a range of features that enable businesses to manage their data with ease.