

**A · P · U**  
ASIA PACIFIC UNIVERSITY  
OF TECHNOLOGY & INNOVATION

## **GROUP ASSIGNMENT (PART2).**

**TECHNOLOGY PARK MALAYSIA.**

**CT042-3-1-IDB.**

**INTRODUCTION TO DATABASE.**

**HAND-OUT DATE: WEEK 3.**

**HAND-IN DATE: WEEK 12.**

**WEIGHTAGE: 60%.**

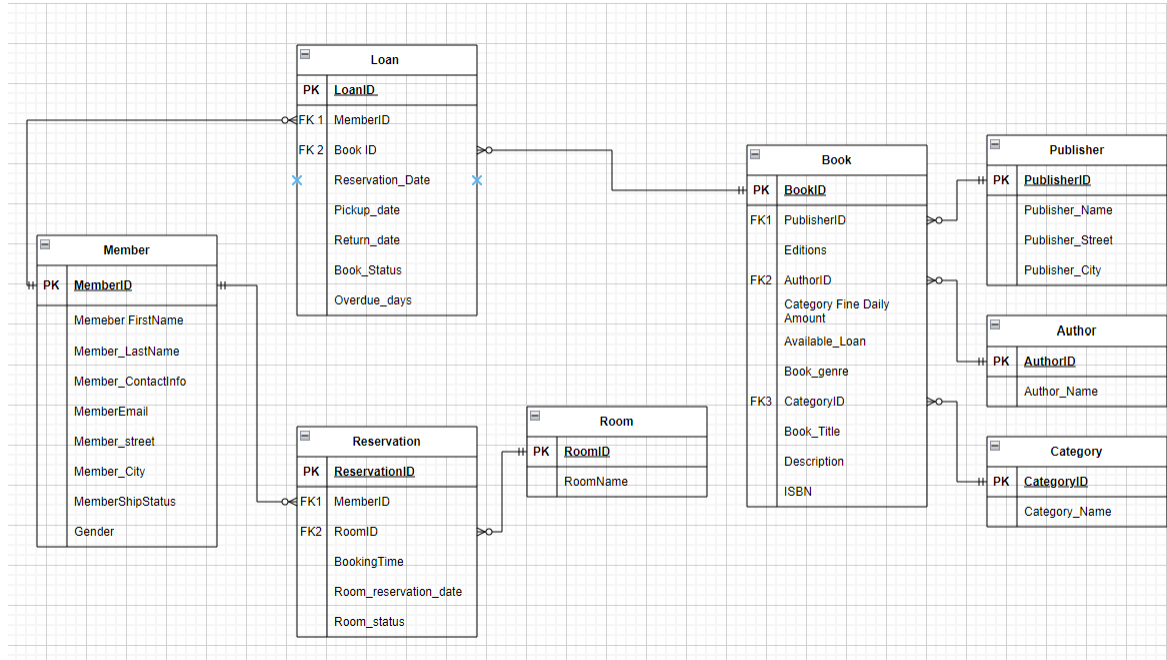
<b><i>STUDENT NAME:</i></b>	<b><i>TP_NUMBER:</i></b>
<i>ANAS WADHAH MOHAMMED QAID</i>	<i>TP077401</i>
<i>COLIN HII ZHI XIANG</i>	<i>TP071461</i>
<i>AHMED ABDULAZIZ OMAR BAFLAH</i>	<i>TP078277</i>
<i>BASEM GHASAN MANSOOR ALI</i>	<i>TP076027</i>

## Contents

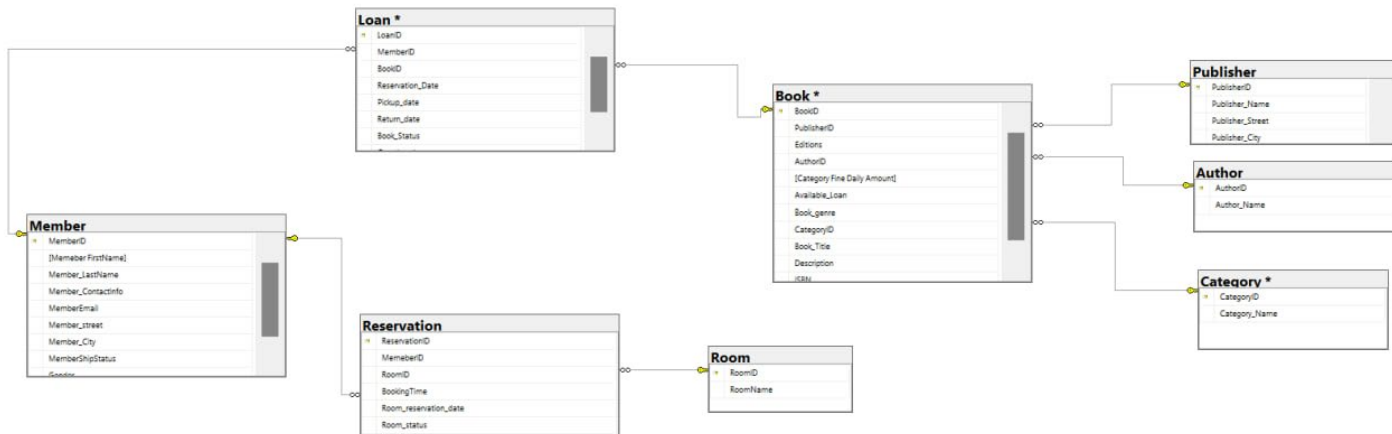
<b>1 Database Schema. ....</b>	<b>3</b>
<b>1.1 Entity Relationship Diagram. ....</b>	<b>3</b>
<b>1.2 Generated Database Diagram from DMBS.....</b>	<b>3</b>
<b>2 SQL- Data Definition Language (DDL). ....</b>	<b>4</b>
<b>2.1 Creating Database &amp; Tables and Insert the Data. ....</b>	<b>Error! Bookmark not defined.</b>
<b>3 SQL-Data Manipulation Language (DML).....</b>	<b>19</b>
<b>3.1 Q1: List all the book names, descriptions, category name and category description that are currently not on loan and available for loan. The results are sorted in alphabetical order of book name.....</b>	<b>19</b>
<b>3.2 Q2: Display the active members and list of books they have borrowed in year 2023. The result must consist of members' first name, last name, book name, and the borrow date. Sort the result by borrow date in ascending order.....</b>	<b>Error!</b>
<b>Bookmark not defined.</b>	
<b>3.3 Q3: Show members who had paid overdue more than 2 times. The results must consist of their first name, last name, contact number, number of overdue, and total due amount.....</b>	<b>Error! Bookmark not defined.</b>
<b>3.4 Q4: Display the name and description of the book borrowed most in each tag color. Sort the result by tag color in descending order.....</b>	<b>22</b>
<b>3.5 Q5: Show the genre which has the greatest number of books.....</b>	<b>23</b>
<b>3.6 Q6: Show the total number of books reserved in year 2023.....</b>	<b>23</b>
<b>3.7 Q7: Show the publisher whose books are least-in-quantity for yellow tagged books.....</b>	<b>14</b>
<b>3.8 Q8: Show the number of books written by each author, sort the result according to book count (author with most books sorted on top).....</b>	<b>25</b>
<b>3.9 Q9: Display the name and contact number of member(s) who had at least borrowed book under genre 'fantasy' once.....</b>	<b>26</b>
<b>3.10 Q10: Show books that cost more than RM50. Result should show name of the book, description, publisher name and publisher address.....</b>	<b>16</b>
<b>4 Work Matrix</b>	<b>28</b>

# 1. Database Schema.

## 1.1 Entity Relationship Diagram.



## 1.2 Generated Database Diagram from DBMS.



## 2 SQL- Data Definition Language (DDL).

### 2.1 Database Dictionary

For database creation, we created 7 tables that follow our initial 3NF design in part 1B. The tables consist of Members, publishers, copies, books, loans, rooms, reservation. We assigned the character limit and datatypes according to the data expected to store in the database, for example: “Member\_Gender” in Members table allow the input of 1 character which is ideal as the data will either be “M” or “F”. Another example is “Description” in Books table, we implemented such a datatype due to the descriptions being usually large in size.

Regarding our foreign keys and primary keys, they will follow the same format that we previously implemented in the 3NF section in part 1B. The following tables will explain in detail.

#### Members

**Member’s table contain the following data:** (MemberID, Member\_FirstName, Member\_LastName, Member\_Gender, Member\_ContactInfo, MemberEmail, Member\_Street, Member\_City, Membership\_Status)

**Primary Key:** MemberID

**Alternate Keys:** NULL

**Foreign Keys:** NULL

**Attributes:**

Name	Description	Data type and length
MemberID	A primary key that uniquely identifies members.	Integer
Member_FirstName	The first name of the member.	Variable character (50)
Member_LastName	The last name of the member.	Variable character (50)
Member_Gender	The member’s gender (either “M” or “F”).	A single character (1)
Member_ContactInfo	The member’s contact information.	Variable character (100)
MemberEmail	Email of member.	Variable character (100)
Member_Street	Member street.	Variable character (100)
Member_City	The city that the member resides in.	Variable character (50)
Membership_Status	Current membership status.	Variable character (20)

### Publishers

**Publisher's table contain the following data:** (PublisherID, Publisher\_Name, Publisher\_Street, Publisher\_City)

**Primary Key:** PublisherID

**Alternate Keys:** NULL

**Foreign Keys:** NULL

**Attributes:**

Name	Description	Data type and length
PublisherID	A primary key that uniquely identifies publishers.	Integer
Publisher_Name	The name of the Publisher.	Variable character (100)
Publisher_Street	Publisher street.	Variable character (100)
Publisher_City	The city that the publisher resides in.	Variable character (50)

### Copies

**Copies' table contain the following data:** (CopiesID, ISBN, Book\_Status)

**Primary Key:** MemberID

**Alternate Keys:** NULL

**Foreign Keys:** NULL

**Attributes:**

Name	Description	Data type and length
CopiesID	A primary key that uniquely identifies book copies.	Integer
ISBN	A numeric commercial book identifier, example: ("978-1400031348")	Variable character (50)
Book_Status	The current status of the book.	Variable character (20)

## Books

**Books's table contain the following data:** (BookID, ISBN, Editions, CopiesID, Author, Category Fine Daily Amount, Available\_Loan, Book\_genre, Book\_category, Book\_Title, PublisherID, Description)

**Primary Key:** BookID

**Alternate Keys:** NULL

**Foreign Keys:** the "PublisherID" and the "CopiesID" columns act as foreign keys, referencing the primary key of the "Publisher" table and the "Copies" tables.

**Attributes:**

Name	Description	Data type and length
BookID	A primary key that uniquely identifies Books.	Integer
ISBN	A numeric commercial book identifier, example: ("978-1400031348")	Variable character (50)
Editions	Identifies the edition of the book	Integer
CopiesID	A foreign key that is linked to the copies table.	Integer
Author	Information about the Author.	Variable character (100)
Category Fine Daily Amount	This gives the daily fine to be paid if the book is overdue	Decimal (10, 2)
Available_Loan	This Tells us if the book is available to loan or not	Variable character (10)
Book_genre	This tells us the genre of the book, whether its horror, romance or comedy.	Variable character (50)
Book_category	This tells us if the book is either yellow-tagged, red-tagged or green-tagged.	Variable character (50)
Book_Title	This stores the book title information	Variable character (100)

PublisherID	A foreign key that is linked to the publisher's table.	Integer
Description	This stores the description of the book.	Text

### Loans

**Member's table contain the following data:** (LoanID, MemberID, BookID, Reservation\_Date, Pickup\_date, Return\_date, Book\_Status, Overdue\_days)

**Primary Key:** LoanID

**Alternate Keys:** NULL

**Foreign Keys:** the "MemberID" and the "BookID" columns serve as foreign keys, referencing the primary keys of the "Members" table and the "Books" tables.

**Attributes:**

Name	Description	Data type and length
LoanID	A primary key that uniquely identifies Loans.	Integer
MemberID	A foreign key that is linked to the Members table.	Integer
BookID	A foreign key that is linked to the Books table.	Integer
Reservation_Date	This attribute stores the date when the member reserved the book. (can be Null)	DATE
Pickup_date	This attribute stores the date when the member picks up the book from the library. (can be Null)	DATE
Return_date	This attribute stores the date when the member returns the book. (can be Null)	DATE
Book_Status	This tells us if the book is available to loan or not	Variable character (20)

Overdue_days	This tells the number of overdue days. (can be Null)	Variable character (20)
--------------	--	-------------------------

### Rooms

**Rooms table contains the following data:** (RoomID, Room\_Name)

**Primary Key:** RoomID

**Alternate Keys:** NULL

**Foreign Keys:** NULL

**Attributes:**

Name	Description	Data type and length
RoomID	A primary key that uniquely identifies the Rooms.	Integer
Room_Name	This slot saves the room names.	Variable character (50)



## Reservations

**Reservations table contain the following data:** (ReservationID, MemberID, RoomID, Booking\_Time, Room\_reservation\_date, Room\_Status)

**Primary Key:** ReservationID

**Alternate Keys:** NULL

**Foreign Keys:** the "MemberID" and the "RoomID" columns serve as foreign keys, referencing the primary keys of the "Members" table and the "Rooms" tables.

**Attributes:**

Name	Description	Data type and length
ReservationID	A primary key that uniquely identifies reservations.	Integer
Booking_Time	The time the reservation was booked.	DATETIME
Room_reservation_date	When the member is going to arrive to the reserved room.	DATE
Room_Status	The status of the room (Booked, Confirmed, etc.)	Variable character (20)
MemberID	A foreign key that is linked to the member's table.	Integer
RoomID	A foreign key that is linked to the room's table.	Integer

## 2.2 Creating Database & Tables

The steps we used to create the database tables involve, creating a database first, then using the database using the command “use ---” which ensures that all commands are directed to the respective database, then creating the tables using the “Create table -----” command as can be seen in the following tables:

```
CREATE DATABASE Database_Assignment;
use Database_Assignment;

-- Create Members Table
CREATE TABLE Members (
    MemberID INT PRIMARY KEY,
    Member_FirstName VARCHAR(50),
    Member_LastName VARCHAR(50),
    Member_Gender CHAR(1),
    Member_ContactInfo VARCHAR(100),
    MemberEmail VARCHAR(100),
    Member_Street VARCHAR(100),
    Member_City VARCHAR(50),
    Membership_Status VARCHAR(20)
);

-- Create Publishers Table
CREATE TABLE Publishers (
    PublisherID INT PRIMARY KEY,
    Publisher_Name VARCHAR(100),
    Publisher_Street VARCHAR(100),
    Publisher_City VARCHAR(50)
);

-- Create Copies Table
CREATE TABLE Copies (
    CopiesID INT PRIMARY KEY,
    ISBN VARCHAR(50),
    Book_Status VARCHAR(20)
);
```

Figure 1: Creation of tables “Members”, “Publishers” and “Copies”

```

-- Create Books Table
CREATE TABLE Books (
    BookID INT PRIMARY KEY,
    ISBN VARCHAR(50),
    Editions INT,
    Author VARCHAR(100),
    Category_Fine_Daily_Amount DECIMAL(10, 2),
    Available_Loan VARCHAR(10),
    Book_genre VARCHAR(50),
    Book_category VARCHAR(50),
    Book_Title VARCHAR(100),
    PublisherID INT,
    Description TEXT,
    CopiesID INT,
    FOREIGN KEY (PublisherID) REFERENCES Publishers(PublisherID),
    FOREIGN KEY (CopiesID) REFERENCES Copies(CopiesID)
);

-- Create Loans Table
CREATE TABLE Loans (
    LoanID INT PRIMARY KEY,
    MemberID INT,
    BookID INT,
    Reservation_Date DATE,
    Pickup_date DATE,
    Return_date DATE,
    Book_Status VARCHAR(20),
    Overdue_days VARCHAR(20),
    FOREIGN KEY (MemberID) REFERENCES Members(MemberID),
    FOREIGN KEY (BookID) REFERENCES Books(BookID)
);

```

Figure 2: Creation of tables “Books” and “Loans”

```

-- Create Rooms Table
CREATE TABLE Rooms (
    RoomID INT PRIMARY KEY,
    Room_Name VARCHAR(50)
);

-- Create Reservations Table
CREATE TABLE Reservations (
    ReservationID INT PRIMARY KEY,
    MemberID INT,
    RoomID INT,
    Booking_Time DATETIME,
    Room_reservation_date DATE,
    Room_Status VARCHAR(20),
    FOREIGN KEY (MemberID) REFERENCES Members(MemberID),
    FOREIGN KEY (RoomID) REFERENCES Rooms(RoomID)
);

```

Figure 3: Creation of tables “Rooms” and “Reservations”

## 2.3 Inserting data into the tables:

The values in the database contain 16 member values and 16 publisher values each with their respective information. The tables also contain data regarding the existing rooms in the library and their availability, it also includes data regarding the information about the current books existing in the library and if they are out on loan or available to be sold. The table was viewed using the command “Select \* from ---” The tables follow a 3NF format such as can be seen below. The Data was added into the tables using “Insert into” command as can be seen in the following tables:

```
-- Insert sample data into Members table
INSERT INTO Members (MemberID, Member_FirstName, Member_LastName, Member_Gender, Member_ContactInfo, MemberEmail, Member_Street, Member_City, Membership_Status)
VALUES
(1, 'John', 'Doe', 'M', '123-456-7890', 'john.doe@example.com', '123 Main St', 'New York', 'Active'),
(2, 'Jane', 'Smith', 'F', '987-654-3210', 'jane.smith@example.com', '456 Elm St', 'Los Angeles', 'Active'),
(3, 'Michael', 'Johnson', 'M', '555-555-5555', 'michael.johnson@example.com', '789 Oak St', 'Chicago', 'Inactive'),
(4, 'Emily', 'Davis', 'F', '222-333-4444', 'emily.davis@example.com', '101 Pine St', 'Houston', 'Active'),
(5, 'Christopher', 'Wilson', 'M', '777-888-9999', 'christopher.wilson@example.com', '202 Maple St', 'Miami', 'Active'),
(6, 'Jessica', 'Martinez', 'F', '333-222-1111', 'jessica.martinez@example.com', '303 Cedar St', 'Seattle', 'Inactive'),
(7, 'David', 'Brown', 'M', '444-555-6666', 'david.brown@example.com', '404 Oak St', 'San Francisco', 'Active'),
(8, 'Sarah', 'Garcia', 'F', '999-888-7777', 'sarah.garcia@example.com', '505 Elm St', 'Dallas', 'Active'),
(9, 'Ahmed', 'Ali', 'M', '1234567890', 'ahmed.ali@gmail.com', '123 Elm Street', 'Cairo', 'Active'),
(10, 'Fatima', 'Hassan', 'F', '2345678901', 'fatima.hassan@gmail.com', '456 Maple Avenue', 'Alexandria', 'Active'),
(11, 'Omar', 'Ibrahim', 'M', '3456789012', 'omar.ibrahim@gmail.com', '789 Oak Lane', 'Giza', 'Inactive'),
(12, 'Layla', 'Mahmoud', 'F', '4567890123', 'layla.mahmoud@gmail.com', '321 Pine Street', 'Shubra', 'Active'),
(13, 'Yousef', 'Khalid', 'M', '5678901234', 'yousef.khalid@gmail.com', '654 Cedar Road', 'Mansoura', 'Inactive'),
(14, 'Amina', 'Said', 'F', '6789012345', 'amina.said@gmail.com', '987 Birch Drive', 'Tanta', 'Active'),
(15, 'Hassan', 'Ahmed', 'M', '7890123456', 'hassan.ahmed@gmail.com', '321 Spruce Way', 'Aswan', 'Inactive'),
(16, 'Noor', 'Mohamed', 'F', '8901234567', 'noor.mohamed@gmail.com', '654 Walnut Street', 'Luxor', 'Active');
```

	MemberID	Member_FirstName	Member_LastName	Member_Gender	Member_ContactInfo	MemberEmail	Member_Street	Member_City	Membership_Status
1	1	John	Doe	M	123-456-7890	john.doe@example.com	123 Main St	New York	Active
2	2	Jane	Smith	F	987-654-3210	jane.smith@example.com	456 Elm St	Los Angeles	Active
3	3	Michael	Johnson	M	555-555-5555	michael.johnson@example.com	789 Oak St	Chicago	Inactive
4	4	Emily	Davis	F	222-333-4444	emily.davis@example.com	101 Pine St	Houston	Active
5	5	Christopher	Wilson	M	777-888-9999	christopher.wilson@example.com	202 Maple St	Miami	Active
6	6	Jessica	Martinez	F	333-222-1111	jessica.martinez@example.com	303 Cedar St	Seattle	Inactive
7	7	David	Brown	M	444-555-6666	david.brown@example.com	404 Oak St	San Francisco	Active
8	8	Sarah	Garcia	F	999-888-7777	sarah.garcia@example.com	505 Elm St	Dallas	Active
9	9	Ahmed	Ali	M	1234567890	ahmed.ali@gmail.com	123 Elm Street	Cairo	Active
10	10	Fatima	Hassan	F	2345678901	fatima.hassan@gmail.com	456 Maple Avenue	Alexandria	Active
11	11	Omar	Ibrahim	M	3456789012	omar.ibrahim@gmail.com	789 Oak Lane	Giza	Inactive
12	12	Layla	Mahmoud	F	4567890123	layla.mahmoud@gmail.com	321 Pine Street	Shubra	Active
13	13	Yousef	Khalid	M	5678901234	yousef.khalid@gmail.com	654 Cedar Road	Mansoura	Inactive
14	14	Amina	Said	F	6789012345	amina.said@gmail.com	987 Birch Drive	Tanta	Active
15	15	Hassan	Ahmed	M	7890123456	hassan.ahmed@gmail.com	321 Spruce Way	Aswan	Inactive
16	16	Noor	Mohamed	F	8901234567	noor.mohamed@gmail.com	654 Walnut Street	Luxor	Active

Figures 4.5: Inserting data into the table “Members” and the outcome

```

-- Insert sample data into Publishers table
INSERT INTO Publishers (PublisherID, Publisher_Name, Publisher_Street, Publisher_City)
VALUES
(1, 'Penguin Random House', '123 Publisher St', 'New York'),
(2, 'HarperCollins', '456 Publisher Ave', 'Los Angeles'),
(3, 'Simon & Schuster', '789 Publisher Blvd', 'Chicago'),
(4, 'Random House', '101 Publisher Blvd', 'Houston'),
(5, 'Macmillan Publishers', '202 Publisher St', 'Miami'),
(6, 'Hachette Livre', '303 Publisher Ave', 'Seattle'),
(7, 'Pearson', '404 Publisher Blvd', 'San Francisco'),
(8, 'Scholastic Corporation', '505 Publisher St', 'Dallas'),
(9, 'Al-Nile Publishing', '12 Salah Salem', 'Cairo'),
(10, 'Al-Maaref', '23 Ramses Street', 'Alexandria'),
(11, 'Dar Al-Hikma', '45 Al-Azhar Street', 'Giza'),
(12, 'Al-Fajr', '67 Tahrir Square', 'Cairo'),
(13, 'Al-Shorouk', '89 Heliopolis', 'Cairo'),
(14, 'Dar Al-Kitab', '101 Nasr City', 'Cairo'),
(15, 'Al-Ahram', '123 El-Galaa Street', 'Alexandria'),
(16, 'Al-Arabi', '145 Mohandessin', 'Giza');

```

	PublisherID	Publisher_Name	Publisher_Street	Publisher_City
1	1	Penguin Random House	123 Publisher St	New York
2	2	HarperCollins	456 Publisher Ave	Los Angeles
3	3	Simon & Schuster	789 Publisher Blvd	Chicago
4	4	Random House	101 Publisher Blvd	Houston
5	5	Macmillan Publishers	202 Publisher St	Miami
6	6	Hachette Livre	303 Publisher Ave	Seattle
7	7	Pearson	404 Publisher Blvd	San Francisco
8	8	Scholastic Corporation	505 Publisher St	Dallas
9	9	Al-Nile Publishing	12 Salah Salem	Cairo
10	10	Al-Maaref	23 Ramses Street	Alexandria
11	11	Dar Al-Hikma	45 Al-Azhar Street	Giza
12	12	Al-Fajr	67 Tahrir Square	Cairo
13	13	Al-Shorouk	89 Heliopolis	Cairo
14	14	Dar Al-Kitab	101 Nasr City	Cairo
15	15	Al-Ahram	123 El-Galaa Street	Alexandria
16	16	Al-Arabi	145 Mohandessin	Giza

Figures 6,7: Inserting data into the table “Publishers” and the outcome



```

-- Insert sample data into Copies table
INSERT INTO Copies (CopiesID, ISBN, Book_Status)
VALUES
(1, '978-0345337665', 'Available'),
(2, '978-0553280365', 'Available'),
(3, '978-0743273565', 'Lent'),
(4, '978-1400031348', 'Available'),
(5, '978-1451673319', 'Lent'),
(6, '978-0679728757', 'Available'),
(7, '978-0141182803', 'Overdue'),
(8, '978-0316769174', 'Overdue'),
(9, '978-3161484100', 'Overdue'),
(10, '978-0596520687', 'Already Reserved'),
(11, '978-1402894626', 'Available'),
(12, '978-0123748560', 'Already Reserved'),
(13, '978-1593275846', 'Available'),
(14, '978-0201633610', 'Lent'),
(15, '978-1449394926', 'Available'),
(16, '978-0262033848', 'Overdue'),
(17, '978-1603090179', 'Lent'),
(18, '978-1603090186', 'Available'),
(19, '978-1603090193', 'Already Reserved'),
(20, '978-1603090209', 'Available'),
(21, '978-1603090216', 'Available'),
(22, '978-1603090223', 'Already Reserved'),
(23, '978-1603090230', 'Lent'),
(24, '978-1603090247', 'Overdue'),
(25, '978-1234567890', 'Available'),
(26, '978-4567890123', 'Lent'),
(27, '978-7890123456', 'Overdue'),
(28, '978-0987654321', 'Available'),
(29, '978-3141592653', 'Already Reserved'),
(30, '978-1617290541', 'Available'),
(31, '978-0071494618', 'Lent');

```

	CopiesID	ISBN	Book_Status
1	1	978-0345337665	Available
2	2	978-0553280365	Available
3	3	978-0743273565	Lent
4	4	978-1400031348	Available
5	5	978-1451673319	Lent
6	6	978-0679728757	Available
7	7	978-0141182803	Overdue
8	8	978-0316769174	Overdue
9	9	978-3161484100	Overdue
10	10	978-0596520687	Already Reserved
11	11	978-1402894626	Available
12	12	978-0123748560	Already Reserved
13	13	978-1593275846	Available
14	14	978-0201633610	Lent
15	15	978-1449394926	Available
16	16	978-0262033848	Overdue
17	17	978-1603090179	Lent
18	18	978-1603090186	Available
19	19	978-1603090193	Already Reserved
20	20	978-1603090209	Available
21	21	978-1603090216	Available
22	22	978-1603090223	Already Reserved
23	23	978-1603090230	Lent
24	24	978-1603090247	Overdue
25	25	978-1234567890	Available
26	26	978-4567890123	Lent
27	27	978-7890123456	Overdue
28	28	978-0987654321	Available
29	29	978-3141592653	Already Reserved
30	30	978-1617290541	Available
31	31	978-0071494618	Lent

Figures 8,9: Inserting data into the table "Copies" and the outcome

```

-- Insert sample data into Rooms table
INSERT INTO Rooms (RoomID, Room_Name)
VALUES
(1, 'Room A'),
(2, 'Room B'),
(3, 'Room C'),
(4, 'Room D'),
(5, 'Room E'),
(6, 'Room F'),
(7, 'Room G'),
(8, 'Room H'),
(9, 'Reading Room A'),
(10, 'Conference Room B'),
(11, 'Study Room C'),
(12, 'Lecture Hall D'),
(13, 'Seminar Room E'),
(14, 'Workshop Room F'),
(15, 'Meeting Room G'),
(16, 'Quiet Room H'),
(17, 'Conference Room A'),
(18, 'Conference Room B'),
(19, 'Study Room 1'),
(20, 'Study Room 2'),
(21, 'Children Room'),
(22, 'Reading Room'),
(23, 'Computer Lab'),
(24, 'Event Hall'),
(25, 'Reading Room 2'),
(26, 'Seminar Room B'),
(27, 'Lecture Hall 1'),
(28, 'Conference Room B'),
(29, 'Study Room 2'),
(30, 'Reading Room 3'),
(31, 'Lecture Hall 2');

```

	RoomID	Room_Name
1	1	Room A
2	2	Room B
3	3	Room C
4	4	Room D
5	5	Room E
6	6	Room F
7	7	Room G
8	8	Room H
9	9	Reading Room A
10	10	Conference Room B
11	11	Study Room C
12	12	Lecture Hall D
13	13	Seminar Room E
14	14	Workshop Room F
15	15	Meeting Room G
16	16	Quiet Room H
17	17	Conference Room A
18	18	Conference Room B
19	19	Study Room 1
20	20	Study Room 2
21	21	Children Room
22	22	Reading Room
23	23	Computer Lab
24	24	Event Hall
25	25	Reading Room 2
26	26	Seminar Room B
27	27	Lecture Hall 1
28	28	Conference Room B
29	29	Study Room 2
30	30	Reading Room 3
31	31	Lecture Hall 2

Figures 10,11: Inserting data into the table “Rooms” and the outcome

```
-- Insert sample data into Books table
INSERT INTO Books (BookID, ISBN, Editions, Author, Category_Fine_Daily_Amount, Available_Loan, Book_genre, Book_category, Book_Title, PublisherID, Description, CopiesID)
VALUES
(1, '978-0345337665', 1, 'J.R.R. Tolkien', 0.50, 'Yes', 'Fantasy', 'Red', 'The Lord of the Rings', 1, 'Epic fantasy novel', 1),
(2, '978-0553280365', 1, 'George Orwell', 0.50, 'Yes', 'Dystopian', 'Yellow', '1984', 2, 'Dystopian novel', 2),
(3, '978-0743273565', 1, 'F. Scott Fitzgerald', 0.50, 'No', 'Classic', 'Red', 'The Great Gatsby', 3, 'American novel', 3),
(4, '978-1408031348', 1, 'Khaled Hosseini', 0.50, 'Yes', 'Historical Fiction', 'Green', 'The Kite Runner', 4, 'Historical novel', 4),
(5, '978-1451673319', 1, 'Ernest Hemingway', 0.50, 'No', 'Classic', 'Red', 'The Old Man and the Sea', 5, 'Novella', 5),
(6, '978-0679728757', 1, 'Gabriel Garcia Marquez', 0.50, 'Yes', 'Magical Realism', 'Green', 'One Hundred Years of Solitude', 6, 'Magic realism novel', 6),
(7, '978-0141182803', 1, 'Leo Tolstoy', 0.50, 'No', 'Historical Fiction', 'Green', 'War and Peace', 7, 'Historical novel', 7),
(8, '978-0316769174', 1, 'J.D. Salinger', 0.50, 'No', 'Coming-of-age', 'Red', 'The Catcher in the Rye', 8, 'Novel', 8),
(9, '978-3161484100', 1, 'Naguib Mahfouz', 1.50, 'No', 'Fiction', 'Yellow', 'Palace Walk', 9, 'A novel set in Cairo.', 9),
(10, '978-0596520687', 2, 'Alaa Al Aswany', 2.00, 'No', 'Fiction', 'Green', 'The Yacoubian Building', 10, 'A story of modern Cairo.', 10),
(11, '978-1402894626', 1, 'Taha Hussein', 1.00, 'Yes', 'Non-Fiction', 'Red', 'The Days', 11, 'An autobiography.', 11),
(12, '978-0123748560', 3, 'Ihsan Abdel Quddous', 1.50, 'No', 'Fiction', 'Green', 'I Do Not Sleep', 12, 'A psychological drama.', 12),
(13, '978-1593275846', 1, 'Sonallah Ibrahim', 2.00, 'Yes', 'Fiction', 'Yellow', 'Zaat', 13, 'A novel about a woman in Cairo.', 13),
(14, '978-0201633610', 2, 'Bahaa Taher', 1.75, 'No', 'Fiction', 'Yellow', 'Sunset Oasis', 14, 'Historical fiction.', 14),
(15, '978-1449394926', 1, 'Ahdaf Soueif', 2.50, 'Yes', 'Fiction', 'Green', 'The Map of Love', 15, 'A love story.', 15),
(16, '978-0262033848', 1, 'Mohamed Choukri', 1.00, 'No', 'Non-Fiction', 'Green', 'For Bread Alone', 16, 'A memoir.', 16),
(17, '978-1603090179', 1, 'Astrid Lindgren', 3.50, 'No', 'Children', 'Red', 'Pippi Longstocking', 3, 'A classic childrens book.', 17),
(18, '978-1603090186', 1, 'Henning Mankell', 4.00, 'Yes', 'Crime', 'Green', 'Faceless Killers', 5, 'A gripping crime novel.', 18),
(19, '978-1603090193', 2, 'Selma Lagerlöf', 2.50, 'No', 'Adventure', 'Red', 'The Wonderful Adventures of Nils', 3, 'An adventurous journey.', 19),
(20, '978-1603090209', 3, 'Karin Boye', 6.00, 'Yes', 'Dystopian', 'Green', 'Kalloccain', 7, 'A dystopian novel.', 20),
(21, '978-1603090216', 1, 'August Strindberg', 3.00, 'Yes', 'Drama', 'Red', 'Miss Julie', 10, 'A classic drama play.', 21),
(22, '978-1603090223', 2, 'Håkan Nesser', 4.50, 'No', 'Thriller', 'Green', 'Borkmann Point', 16, 'A thrilling detective story.', 22),
(23, '978-1603090230', 3, 'Camilla Läckberg', 5.50, 'No', 'Crime', 'Red', 'The Ice Princess', 1, 'A captivating crime novel.', 23),
(24, '978-1603090247', 1, 'Maj Sjöwall', 3.75, 'No', 'Crime', 'Yellow', 'Roseanna', 9, 'A foundational crime story.', 24),
(25, '978-1234567890', 2, 'Clarice Lispector', 1.75, 'Yes', 'Fiction', 'Yellow', 'Family Ties', 12, 'A collection of short stories.', 25),
(26, '978-4567890123', 1, 'Machado de Assis', 2.25, 'No', 'Fiction', 'Green', 'Dom Casmurro', 2, 'A novel about love and betrayal.', 26),
(27, '978-7890123456', 4, 'Graciliano Ramos', 1.50, 'No', 'Fiction', 'Green', 'Barren Lives', 6, 'A story of a poor family in the Brazilian outback.', 27),
(28, '978-9987654321', 3, 'Rubem Fonseca', 2.00, 'Yes', 'Fiction', 'Red', 'The Taker and Other Stories', 8, 'A collection of crime stories.', 28),
(29, '978-3141592653', 1, 'Luis Fernando Verissimo', 1.75, 'No', 'Humor', 'Red', 'The Club of Angels', 11, 'A dark comedy about a dinner club.', 29),
(30, '978-1617290541', 2, 'Nelson Rodrigues', 2.25, 'Yes', 'Drama', 'Green', 'All Nudity Shall Be Punished', 3, 'A controversial play.', 30),
(31, '978-0071494618', 1, 'Raduan Nassar', 1.50, 'No', 'Fiction', 'Green', 'Ancient Tillage', 5, 'A story about a young man returning to his rural home.', 31);
```

BookID	ISBN	Editions	Author	Category_Fine_Daily_Amount	Available_Loan	Book_genre	Book_category	Book_Title	PublisherID	Description	CopiesID
1	978-0345337665	1	J.R.R. Tolkien	0.50	Yes	Fantasy	Red	The Lord of the Rings	1	Epic fantasy novel	1
2	978-0553280365	1	George Orwell	0.50	Yes	Dystopian	Yellow	1984	2	Dystopian novel	2
3	978-0743273565	1	F. Scott Fitzgerald	0.50	No	Classic	Red	The Great Gatsby	3	American novel	3
4	978-1408031348	1	Khaled Hosseini	0.50	Yes	Historical Fiction	Green	The Kite Runner	4	Historical novel	4
5	978-1451673319	1	Ernest Hemingway	0.50	No	Classic	Red	The Old Man and the Sea	5	Novella	5
6	978-0679728757	1	Gabriel Garcia Marquez	0.50	Yes	Magical Realism	Green	One Hundred Years of Solitude	6	Magic realism novel	6
7	978-0141182803	1	Leo Tolstoy	0.50	No	Historical Fiction	Green	War and Peace	7	Historical novel	7
8	978-0316769174	1	J.D. Salinger	0.50	No	Coming-of-age	Red	The Catcher in the Rye	8	Novel	8
9	978-3161484100	1	Naguib Mahfouz	1.50	No	Fiction	Yellow	Palace Walk	9	A novel set in Cairo.	9
10	978-0596520687	2	Alaa Al Aswany	2.00	No	Fiction	Green	The Yacoubian Building	10	A story of modern Cairo.	10
11	978-1402894626	1	Taha Hussein	1.00	Yes	Non-Fiction	Red	The Days	11	An autobiography.	11
12	978-0123748560	3	Ihsan Abdel Quddous	1.50	No	Fiction	Green	I Do Not Sleep	12	A psychological drama.	12
13	978-1593275846	1	Sonallah Ibrahim	2.00	Yes	Fiction	Yellow	Zaat	13	A novel about a woma...	13
14	978-0201633610	2	Bahaa Taher	1.75	No	Fiction	Yellow	Sunset Oasis	14	Historical fiction.	14
15	978-1449394926	1	Ahdaf Soueif	2.50	Yes	Fiction	Green	The Map of Love	15	A love story.	15
16	978-0262033848	1	Mohamed Choukri	1.00	No	Non-Fiction	Green	For Bread Alone	16	A memoir.	16
17	978-1603090179	1	Astrid Lindgren	3.50	No	Children	Red	Pippi Longstocking	3	A classic childrens bo...	17
18	978-1603090186	1	Henning Mankell	4.00	Yes	Crime	Green	Faceless Killers	5	A gripping crime novel.	18
19	978-1603090193	2	Selma Lagerlöf	2.50	No	Adventure	Red	The Wonderful Adventures of ...	3	An adventurous journey.	19
20	978-1603090209	3	Karin Boye	6.00	Yes	Dystopian	Green	Kalloccain	7	A dystopian novel.	20
21	978-1603090216	1	August Strindberg	3.00	Yes	Drama	Red	Miss Julie	10	A classic drama play.	21
22	978-1603090223	2	Håkan Nesser	4.50	No	Thriller	Green	Borkmann Point	16	A thrilling detective sto...	22
23	978-1603090230	3	Camilla Läckberg	5.50	No	Crime	Red	The Ice Princess	1	A captivating crime no...	23
24	978-1603090247	1	Maj Sjöwall	3.75	No	Crime	Yellow	Roseanna	9	A foundational crime s...	24
25	978-1234567890	2	Clarice Lispector	1.75	Yes	Fiction	Yellow	Family Ties	12	A collection of short st...	25
26	978-4567890123	1	Machado de Assis	2.25	No	Fiction	Green	Dom Casmurro	2	A novel about love an...	26
27	978-7890123456	4	Graciliano Ramos	1.50	No	Fiction	Green	Barren Lives	6	A story of a poor family...	27
28	978-9987654321	3	Rubem Fonseca	2.00	Yes	Fiction	Red	The Taker and Other Stories	8	A collection of crime st...	28
29	978-3141592653	1	Luis Fernando Verissi...	1.75	No	Humor	Red	The Club of Angels	11	A dark comedy about ...	29
30	978-1617290541	2	Nelson Rodrigues	2.25	Yes	Drama	Green	All Nudity Shall Be Punished	3	A controversial play.	30
31	978-0071494618	1	Raduan Nassar	1.50	No	Fiction	Green	Ancient Tillage	5	A story about a young ...	31

Figures 12,13: Inserting data into the table "Books" and the outcome



```

-- Insert sample data into Loans table
INSERT INTO Loans (LoanID, MemberID, BookID, Reservation_Date, Pickup_date, Return_date, Book_Status, Overdue_days)
VALUES
(1, 1, 1, NULL, NULL, NULL, 'Available', NULL),
(2, 2, 2, NULL, NULL, NULL, 'Available', NULL),
(3, 3, 3, '2023-06-03', '2023-06-05', '2023-07-05', 'Lent', NULL),
(4, 4, 4, NULL, NULL, NULL, 'Available', NULL),
(5, 5, 5, '2023-06-05', '2023-06-08', '2023-07-08', 'Lent', NULL),
(6, 6, 6, NULL, NULL, NULL, 'Available', NULL),
(7, 7, 7, '2023-06-05', '2023-06-08', '2023-06-23', 'Overdue', '10'),
(8, 8, 8, '2023-06-08', '2023-06-10', '2023-07-10', 'Overdue', '32'),
(9, 9, 9, '2023-06-05', '2023-06-20', '2023-06-10', 'Overdue', '20'),
(10, 10, 10, '2023-07-03', '2023-07-18', '2023-08-02', 'Already Reserved', NULL),
(11, 11, 11, NULL, NULL, NULL, 'Available', NULL),
(12, 12, 12, '2024-07-07', '2024-07-10', '2024-07-25', 'Already Reserved', NULL),
(13, 13, 13, NULL, NULL, NULL, 'Available', NULL),
(14, 14, 14, '2024-07-11', '2024-07-12', '2024-07-22', 'Lent', NULL),
(15, 15, 15, NULL, NULL, NULL, 'Available', NULL),
(16, 16, 16, '2024-06-05', '2024-06-20', '2024-07-05', 'Overdue', '7'),
(17, 1, 17, '2025-05-03', '2025-05-04', '2025-06-03', 'Lent', NULL),
(18, 2, 18, NULL, NULL, NULL, 'Available', NULL),
(19, 3, 19, '2025-05-07', '2025-05-08', '2025-06-07', 'Already Reserved', NULL),
(20, 4, 20, NULL, NULL, NULL, 'Available', NULL),
(21, 5, 21, NULL, NULL, NULL, 'Available', NULL),
(22, 6, 22, '2025-05-13', '2025-05-14', '2025-05-19', 'Already Reserved', NULL),
(23, 7, 23, '2025-05-15', '2025-05-16', '2025-06-15', 'Lent', NULL),
(24, 8, 24, '2025-06-05', '2025-06-20', '2025-07-10', 'Overdue', '22'),
(25, 9, 25, '2025-06-05', '2025-06-20', '2025-07-10', 'Overdue', '28'),
(26, 10, 26, '2025-07-18', '2025-07-20', '2025-08-04', 'Lent', NULL),
(27, 11, 27, '2025-07-19', '2025-07-22', '2025-08-06', 'Overdue', '4'),
(28, 12, 28, NULL, NULL, NULL, 'Available', NULL),
(29, 13, 29, '2025-07-21', '2025-07-23', '2025-08-22', 'Already Reserved', NULL),
(30, 14, 30, NULL, NULL, NULL, 'Available', NULL),
(31, 15, 31, '2025-07-23', '2025-07-25', '2025-08-09', 'Lent', NULL);

```

	LoanID	MemberID	BookID	Reservation_Date	Pickup_date	Return_date	Book_Status	Overdue_days
1	1	1	1	NULL	NULL	NULL	Available	NULL
2	2	2	2	NULL	NULL	NULL	Available	NULL
3	3	3	3	2023-06-03	2023-06-05	2023-07-05	Lent	NULL
4	4	4	4	NULL	NULL	NULL	Available	NULL
5	5	5	5	2023-06-05	2023-06-08	2023-07-08	Lent	NULL
6	6	6	6	NULL	NULL	NULL	Available	NULL
7	7	7	7	2023-06-05	2023-06-08	2023-06-23	Overdue	10
8	8	8	8	2023-06-08	2023-06-10	2023-07-10	Overdue	32
9	9	9	9	2023-06-05	2023-06-20	2023-06-10	Overdue	20
10	10	10	10	2023-07-03	2023-07-18	2023-08-02	Already Reserved	NULL
11	11	11	11	NULL	NULL	NULL	Available	NULL
12	12	12	12	2024-07-07	2024-07-10	2024-07-25	Already Reserved	NULL
13	13	13	13	NULL	NULL	NULL	Available	NULL
14	14	14	14	2024-07-11	2024-07-12	2024-07-22	Lent	NULL
15	15	15	15	NULL	NULL	NULL	Available	NULL
16	16	16	16	2024-06-05	2024-06-20	2024-07-05	Overdue	7
17	17	1	17	2025-05-03	2025-05-04	2025-06-03	Lent	NULL
18	18	2	18	NULL	NULL	NULL	Available	NULL
19	19	3	19	2025-05-07	2025-05-08	2025-06-07	Already Reserved	NULL
20	20	4	20	NULL	NULL	NULL	Available	NULL
21	21	5	21	NULL	NULL	NULL	Available	NULL
22	22	6	22	2025-05-13	2025-05-14	2025-05-19	Already Reserved	NULL
23	23	7	23	2025-05-15	2025-05-16	2025-06-15	Lent	NULL
24	24	8	24	2025-06-05	2025-06-20	2025-07-10	Overdue	22
25	25	9	25	2025-06-05	2025-06-20	2025-07-10	Overdue	28
26	26	10	26	2025-07-18	2025-07-20	2025-08-04	Lent	NULL
27	27	11	27	2025-07-19	2025-07-22	2025-08-06	Overdue	4
28	28	12	28	NULL	NULL	NULL	Available	NULL
29	29	13	29	2025-07-21	2025-07-23	2025-08-22	Already Reserved	NULL
30	30	14	30	NULL	NULL	NULL	Available	NULL
31	31	15	31	2025-07-23	2025-07-25	2025-08-09	Lent	NULL

Figures 14,15: Inserting data into the table "Loans" and the outcome

```

-- Insert sample data into Reservations table
INSERT INTO Reservations (ReservationID, MemberID, RoomID, Booking_Time, Room_reservation_date, Room_Status)
VALUES
(1, 1, 1, '2024-06-05 10:00:00', '2024-06-05', 'Pending'),
(2, 2, 2, '2024-06-06 11:00:00', '2024-06-06', 'Confirmed'),
(3, 3, 3, '2024-06-07 12:00:00', '2024-06-07', 'Confirmed'),
(4, 4, 4, '2024-06-08 13:00:00', '2024-06-08', 'Pending'),
(5, 5, 5, '2024-06-09 14:00:00', '2024-06-09', 'Confirmed'),
(6, 6, 6, '2024-06-10 15:00:00', '2024-06-10', 'Cancelled'),
(7, 7, 7, '2024-06-11 16:00:00', '2024-06-11', 'Confirmed'),
(8, 8, 8, '2024-06-12 17:00:00', '2024-06-12', 'Confirmed'),
(9, 9, 9, '2024-01-01 10:00:00', '2024-01-02', 'Confirmed'),
(10, 10, 10, '2024-01-03 11:00:00', '2024-01-04', 'Confirmed'),
(11, 11, 11, '2024-01-05 12:00:00', '2024-01-06', 'Cancelled'),
(12, 12, 12, '2024-01-07 13:00:00', '2024-01-08', 'Confirmed'),
(13, 13, 13, '2024-01-09 14:00:00', '2024-01-10', 'Pending'),
(14, 14, 14, '2024-01-11 15:00:00', '2024-01-12', 'Confirmed'),
(15, 15, 15, '2024-01-13 16:00:00', '2024-01-14', 'Cancelled'),
(16, 16, 16, '2024-01-15 17:00:00', '2024-01-16', 'Confirmed'),
(17, 1, 17, '2023-06-02 11:00:00', '2023-06-03', 'Confirmed'),
(18, 2, 18, '2023-06-03 12:00:00', '2023-06-04', 'Confirmed'),
(19, 3, 19, '2023-06-04 13:00:00', '2023-06-05', 'Confirmed'),
(20, 4, 20, '2023-06-05 14:00:00', '2023-06-06', 'Confirmed'),
(21, 5, 21, '2023-06-06 15:00:00', '2023-06-07', 'Confirmed'),
(22, 6, 22, '2023-06-07 16:00:00', '2023-06-08', 'Confirmed'),
(23, 7, 23, '2023-06-08 17:00:00', '2023-06-09', 'Confirmed'),
(24, 8, 24, '2023-06-09 18:00:00', '2023-06-10', 'Confirmed'),
(25, 9, 25, '2024-01-17 12:00:00', '2024-01-17', 'Cancelled'),
(26, 10, 26, '2024-01-18 13:00:00', '2024-01-18', 'Confirmed'),
(27, 11, 27, '2024-01-19 14:00:00', '2024-01-19', 'Confirmed'),
(28, 12, 28, '2024-01-20 15:00:00', '2024-01-20', 'Confirmed'),
(29, 13, 29, '2024-01-21 16:00:00', '2024-01-21', 'Confirmed'),
(30, 14, 30, '2024-01-22 17:00:00', '2024-01-22', 'Confirmed'),
(31, 15, 31, '2024-01-23 18:00:00', '2024-01-23', 'Cancelled');

```

	ReservationID	MemberID	RoomID	Booking_Time	Room_reservation_date	Room_Status
1	1	1	1	2024-06-05 10:00:00.000	2024-06-05	Pending
2	2	2	2	2024-06-06 11:00:00.000	2024-06-06	Confirmed
3	3	3	3	2024-06-07 12:00:00.000	2024-06-07	Confirmed
4	4	4	4	2024-06-08 13:00:00.000	2024-06-08	Pending
5	5	5	5	2024-06-09 14:00:00.000	2024-06-09	Confirmed
6	6	6	6	2024-06-10 15:00:00.000	2024-06-10	Cancelled
7	7	7	7	2024-06-11 16:00:00.000	2024-06-11	Confirmed
8	8	8	8	2024-06-12 17:00:00.000	2024-06-12	Confirmed
9	9	9	9	2024-01-01 10:00:00.000	2024-01-02	Confirmed
10	10	10	10	2024-01-03 11:00:00.000	2024-01-04	Confirmed
11	11	11	11	2024-01-05 12:00:00.000	2024-01-06	Cancelled
12	12	12	12	2024-01-07 13:00:00.000	2024-01-08	Confirmed
13	13	13	13	2024-01-09 14:00:00.000	2024-01-10	Pending
14	14	14	14	2024-01-11 15:00:00.000	2024-01-12	Confirmed
15	15	15	15	2024-01-13 16:00:00.000	2024-01-14	Cancelled
16	16	16	16	2024-01-15 17:00:00.000	2024-01-16	Confirmed
17	17	1	17	2023-06-02 11:00:00.000	2023-06-03	Confirmed
18	18	2	18	2023-06-03 12:00:00.000	2023-06-04	Confirmed
19	19	3	19	2023-06-04 13:00:00.000	2023-06-05	Confirmed
20	20	4	20	2023-06-05 14:00:00.000	2023-06-06	Confirmed
21	21	5	21	2023-06-06 15:00:00.000	2023-06-07	Confirmed
22	22	6	22	2023-06-07 16:00:00.000	2023-06-08	Confirmed
23	23	7	23	2023-06-08 17:00:00.000	2023-06-09	Confirmed
24	24	8	24	2023-06-09 18:00:00.000	2023-06-10	Confirmed
25	25	9	25	2024-01-17 12:00:00.000	2024-01-17	Cancelled
26	26	10	26	2024-01-18 13:00:00.000	2024-01-18	Confirmed
27	27	11	27	2024-01-19 14:00:00.000	2024-01-19	Confirmed
28	28	12	28	2024-01-20 15:00:00.000	2024-01-20	Confirmed
29	29	13	29	2024-01-21 16:00:00.000	2024-01-21	Confirmed
30	30	14	30	2024-01-22 17:00:00.000	2024-01-22	Confirmed
31	31	15	31	2024-01-23 18:00:00.000	2024-01-23	Cancelled

Figures 16,17: Inserting data into the table “Reservations” and the outcome

## 3 SQL-Data Manipulation Language (DML).

**3.1 Q1: List all the book names, descriptions, category name and category description that are currently not on loan and available for loan. The results are sorted in alphabetical order of book name.**

```
-- This query lists available books with their details and category information.
SELECT
  Books.Book_Title AS BookName,
  Books.Description AS BookDescription,
  Books.Book_category AS CategoryName,
  Books.Book_genre AS CategoryDescription
FROM
  Books
JOIN
  Copies ON Books.CopiesID = Copies.CopiesID
WHERE
  Copies.Book_Status = 'Available'
AND Books.Available_Loan = 'Yes'
ORDER BY
  Books.Book_Title ASC;
```

	BookName	BookDescription	CategoryName	CategoryDescription
1	1984	Dystopian novel	Yellow	Dystopian
2	All Nudity Shall Be Punished	A controversial play.	Green	Drama
3	Barren Lives	A story of a poor family in the Brazilian outback.	Green	Fiction
4	Faceless Killers	A gripping crime novel.	Green	Crime
5	Family Ties	A collection of short stories.	Yellow	Fiction
6	For Bread Alone	A memoir.	Green	Non-Fiction
7	Kallocain	A dystopian novel.	Green	Dystopian
8	Miss Julie	A classic drama play.	Red	Drama
9	One Hundred Years of Soli...	Magic realism novel	Green	Magical Realism
10	Palace Walk	A novel set in Cairo.	Yellow	Fiction
11	Roseanna	A foundational crime story.	Yellow	Crime
12	The Days	An autobiography.	Red	Non-Fiction
13	The Kite Runner	Historical novel	Green	Historical Fiction
14	The Lord of the Rings	Epic fantasy novel	Red	Fantasy
15	The Map of Love	A love story.	Green	Fiction
16	The Taker and Other Stories	A collection of crime stories.	Red	Fiction
17	War and Peace	Historical novel	Green	Historical Fiction
18	Zaat	A novel about a woman in Cairo.	Yellow	Fiction

This query retrieves the titles, descriptions, category names, and category descriptions of books that are currently available for loan and not on loan. It selects these fields from the Books table and joins with the 'Copies' table to check the book status. The 'WHERE' clause ensures that only books marked as available for loan and not currently on loan are included. Finally, the results are sorted alphabetically by book title.

### 3.2 Q2: Display the active members and list of books they have borrowed in year 2023. The result must consist of members' first name, last name, book name, and the borrow date. Sort the result by borrow date in ascending order.

```
-- Retrieve active members' borrowed books in 2023, ordered by borrow date
SELECT
  m.Member_FirstName,
  m.Member_LastName,
  b.Book_Title AS Book_Name,
  l.Pickup_date AS Borrow_Date
FROM
  Members m
JOIN
  Loans l ON m.MemberID = l.MemberID
JOIN
  Books b ON l.BookID = b.BookID
WHERE
  m.Membership_Status = 'Active' AND
  YEAR(l.Pickup_date) = 2023
ORDER BY
  l.Pickup_date ASC;
```

	Member_FirstName	Member_LastName	Book_Name	Borrow_Date
1	Emma	Larsson	Pippi Longstocking	2023-05-04
2	Lars	Karlsson	Faceless Killers	2023-05-06
3	Hanna	Nilsson	The Wonderful Adventures of Nils	2023-05-08
4	Sara	Olsson	Miss Julie	2023-05-12
5	Anna	Svensson	The Ice Princess	2023-05-16

This query retrieves a list of active members who borrowed books in 2023, including their first and last names, the title of the book, and the borrow date. The results are ordered by the borrow date in ascending order. The query joins the **'Members'**, **'Loans'**, and **'Books'** tables based on **'MemberID'** and **'BookID'**.



### 3.3 Q3: Show members who had paid overdue more than 2 times. The results must consist of their first name, last name, contact number, number of overdue, and total due amount.

```
--show members who had paid overdue more than 2 times
SELECT
    M.Member_FirstName,
    M.Member_LastName,
    M.Member_ContactInfo,
    COUNT(L.LoanID) AS Number_of_overdue,
    SUM(B.Category_Fine_Daily_Amount * DATEDIFF(DAY, L.Pickup_date, GETDATE())) AS Total_due_amount
FROM
    Members M
INNER JOIN Loans L ON M.MemberID = L.MemberID
INNER JOIN Books B ON L.BookID = B.BookID
WHERE
    L.Book_Status = 'Overdue'
GROUP BY
    M.MemberID, M.Member_FirstName, M.Member_LastName, M.Member_ContactInfo;
```

Results		Messages			
	Member_FirstName	Member_LastName	Member_ContactInfo	Number_of_overdue	Total_due_amount
1	Yousef	Khalid	5678901234	1	302.00

This query retrieves the first name, last name, and contact information of members who have had books overdue more than two times. It counts the number of overdue books for each member and filters the result to only include members with more than two overdue books.

### 3.4 Q4: Display the name and description of the book borrowed most in each tag color. Sort the result by tag color in descending order.

```
-- Find the most borrowed book in each category
WITH MostBorrowedBooks AS (
SELECT
    b.Book_category,
    b.Book_Title,
    CAST(b.Description AS VARCHAR(MAX)) AS Description,
    COUNT(l.LoanID) AS Borrow_Count,
    ROW_NUMBER() OVER (PARTITION BY b.Book_category ORDER BY COUNT(l.LoanID) DESC) AS rn
FROM
    Books b
JOIN
    Loans l ON b.BookID = l.BookID
GROUP BY
    b.Book_category, b.Book_Title, CAST(b.Description AS VARCHAR(MAX))
)
SELECT
    Book_category,
    Book_Title,
    Description
FROM
    MostBorrowedBooks
WHERE
    rn = 1
ORDER BY
    Book_category DESC;
```

	Book_category	Book_Title	Description
1	Short Stories	Family Ties	A collection of short stories.
2	Play	All Nudity Shall Be Punished	A controversial play.
3	Novel	Ancient Tillage	A story about a young man returning to his rural...
4	Memoir	For Bread Alone	A memoir.
5	Fiction	1984	Dystopian novel
6	Biography	The Days	An autobiography.

This query uses a Common Table Expression (CTE) to find the most borrowed book in each tag color. It counts the number of loans for each book and uses 'ROW\_NUMBER' to rank the books within each category. The outer query selects the top-ranked (most borrowed) book in each category and sorts the results by tag color in descending order.

### 3.5 Q5: Show the genre which has the greatest number of books.

```
-- Find the genre with the highest number of books.
SELECT TOP 1 Book_genre AS Genre, COUNT(*) AS NumBooks
FROM Books
GROUP BY Book_genre
ORDER BY COUNT(*) DESC;
```

	Genre	NumBooks
1	Fiction	11

This query aims to identify the genre with the highest number of books in the database. It achieves this by counting the number of books for each genre using the **'COUNT (\*)'** function along with grouping by the **'Book\_genre'** column. The **'TOP 1'** clause ensures that only the genre with the highest count is returned. Finally, the results are sorted in descending order of book count using **'ORDER BY 'COUNT (\*)' DESC'**.

### 3.6 Q6: Show the total number of books reserved in year 2023.

```
-- This query calculates the total number of books reserved in 2023.
SELECT
    COUNT(BookID) AS TotalReservedBooks
FROM
    Loans
WHERE
    YEAR(Reservation_Date) = 2023;
```

	TotalReservedBooks
1	8

This query calculates the total number of books reserved in the year 2023. It counts the distinct book IDs from the **'Loans'** table, joined with the **'Books'** table, where the reservation date falls within the year 2023. This ensures that only reservations made in 2023 are counted.

### 3.7 Q7: Show the publisher whose books are least-in-quantity for yellow tagged books.

```
-- Identify the publisher with the fewest yellow tagged books.
SELECT TOP 1 P.Publisher_Name, COUNT(*) AS NumYellowBooks
FROM Books AS B
JOIN Publishers AS P ON B.PublisherID = P.PublisherID
WHERE B.Book_genre = 'Yellow'
GROUP BY P.Publisher_Name
ORDER BY COUNT(*) ASC;
```

Results		Messages	
Publisher_Name	NumYellowBooks		

This query is designed to find the publisher who has the lowest number of books with a yellow tag in the database. It starts by joining the **Books** and **Publishers** tables on the **PublisherID** column to link each book with its publisher. Then, it filters the books to include only those with the genre Yellow. Next, it groups the results by the publishers name and counts the number of books for each publisher. **The TOP 1** clause ensures that only the publisher with the fewest yellow-tagged books is returned. Finally, the results are sorted in ascending order of the count of yellow-tagged books using **ORDER BY 'COUNT (\*)' ASC**.



### 3.8 Q8: Show the number of books written by each author, sort the result according to book count (author with most books sorted on top).

-- This query shows the number of books written by each author.

```
SELECT
    Books.Author,
    COUNT(Books.BookID) AS NumberOfBooks
FROM
    Books
GROUP BY
    Books.Author
ORDER BY
    NumberOfBooks DESC;
```

	Author	NumberOfBooks
1	Ahdaf Soueif	1
2	Alaa Al Aswany	1
3	Astrid Lindgren	1
4	August Strindberg	1
5	Bahaa Taher	1
6	Camilla Läckberg	1
7	Clarice Lispector	1
8	Ernest Hemingway	1
9	F. Scott Fitzgerald	1
10	Gabriel Garcia M...	1
11	George Orwell	1
12	Graciliano Ramos	1
13	Håkan Nesser	1
14	Henning Mankell	1
15	Ihsan Abdel Qud...	1
16	J.D. Salinger	1
17	J.R.R. Tolkien	1
18	Karin Boye	1
19	Khaled Hosseini	1
20	Leo Tolstoy	1
21	Luis Fernando Ve...	1
22	Machado de Assis	1
23	Maj Sjöwall	1
24	Mohamed Choukri	1
25	Naguib Mahfouz	1
26	Nelson Rodrigues	1
27	Raduan Nassar	1
28	Rubem Fonseca	1
29	Selma Lagerlöf	1
30	Sonallah Ibrahim	1
31	Taha Hussein	1

This query displays the number of books written by each author, sorted by the book count in descending order. It selects the author and counts the number of books for each author from the 'Books' table, grouping by author. The results are then ordered by the number of books each author has written.

### 3.9 Q9: Display the name and contact number of member(s) who had at least borrowed book under genre ‘fantasy’ once.

```
-- This query lists members who have borrowed books in the fantasy genre.  
SELECT DISTINCT  
    m.Member_FirstName,  
    m.Member_LastName,  
    m.Member_ContactInfo AS ContactNumber  
FROM  
    Loans l  
JOIN  
    Members m ON l.MemberID = m.MemberID  
JOIN  
    Books b ON l.BookID = b.BookID  
WHERE  
    b.Book_genre = 'Fantasy';
```

	Member_FirstName	Member_LastName	ContactNumber
1	John	Doe	123-456-7890

This query lists the names and contact numbers of members who have borrowed at least one book in the fantasy genre. It selects distinct member names and contact details by joining the ‘Loans’, ‘Members’, and ‘Books’ tables, with a ‘WHERE’ clause filtering for books in the fantasy genre.

### 3.10 Q10: Show books that cost more than RM50. Result should show name of the book, description, publisher name and publisher address.

```
-- This query retrieves books costing more than RM50 with their publisher details.
SELECT
  b.Book_Title AS BookName,
  b.Description AS BookDescription,
  p.Publisher_Name,
  CONCAT(p.Publisher_Street, ', ', p.Publisher_City) AS PublisherAddress
FROM
  Books b
JOIN
  Publishers p ON b.PublisherID = p.PublisherID
WHERE
  b.Category_Fine_Daily_Amount > 50;
```

Results		Messages	
BookName	BookDescription	Publisher_Name	PublisherAddress

This query retrieves books that cost more than RM50, displaying the book name, description, publisher name, and publisher address. It joins the **'Books'** table with the **'Publishers'** table and uses a **'WHERE'** clause to filter books whose daily fine amount exceeds RM50. The selected fields include the book title, description, publisher name, and address details.

## Work Matrix.

<b>Part</b>	<b>Component</b>	<b>ANAS WADHAH MOHAMMED QAID</b>	<b>COLIN HII ZHI XIANG</b>	<b>AHMED ABDULAZIZ OMAR BAFLAH</b>	<b>BASEM GHASAN MANSOOR ALI</b>	<b>Total</b>
<b>2</b>	d) Database Schema		<b>50%</b>		<b>50%</b>	<b>100%</b>
<b>2</b>	e) SQL-Data Definition Language (DDL)	<b>25%</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>	<b>100%</b>
<b>2</b>	f) SQL-Data Manipulation Language (DML)	<b>50%</b>		<b>50%</b>		<b>100%</b>