MSc in Computing and Information Systems

Library Database Systems Coursework

December 1st 2023

Group 7
LI-HSUAN LIN
RIKUTO MATSUKURA
WEI-HSUAN HUANG
ZHIWEN QIU

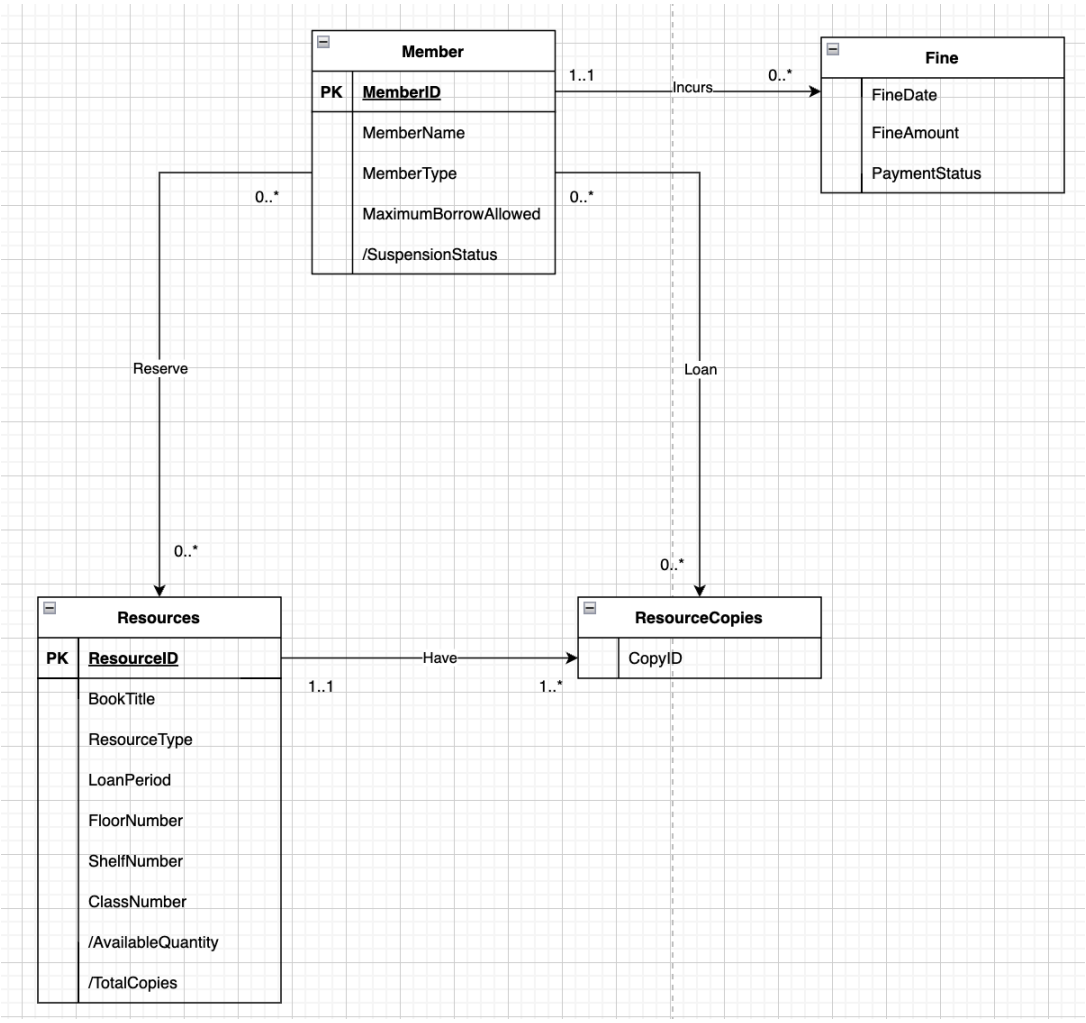
Table of Contents

Assumptions

- For resources that can only be used in-library, we show 0 days in the loan period.
- The due date calculation for resources on loan is based on the loan period.
- All Loan, Return, and Reservation dates fall within the period from 10/1 to 11/30 (it is easy to review).
- Three Loan statuses: “Checked Out”, “Overdue”, and “Returned.” Checked Out: the resource is currently borrowed within the loan period. Overdue: the resource is borrowed but has not been returned after the due date. In our system, we maintain a record of Returned resources in the Loan table for one week. If the date exceeds one week, the record will be moved to the historical records.
- Four Reservation statuses: “Waitlisted”, “Completed”, “Notified”, and “Cancelled.”
Waitlisted: a member is awaiting availability since the resource is currently in use.
Completed: a member successfully acquires the reserved resources within three attempts.
Notified: the reserved resource is available, but the member has not yet collected it.
Cancelled: a member does not take up the reserved resource within three attempts.
- The "RemainingChances" attribute in the Reservation entity is used to track the number of opportunities a member has left to take up the reserved resource.
- The LoanHistory entity is established to prevent an excessive number of records in the Loan entity. After resources have been returned to the library over a week, the borrowing and returning record is archived in LoanHistory.
- The fines accrued by a member on the same date will be summed up and recorded in the Fine table.

Part 1: DATABASE DESIGN

1a) Conceptual Schema



Member:

Explanation: This entity records information about library members. MemberID is the primary key, meaning each member has a unique ID. MemberName stores the member's name, MemberType indicates the member's type (e.g., student, staff), and MaximumBorrowAllowed represents the maximum number of items a member can borrow. The derived attribute /SuspensionStatus can be calculated from the “FineAmount” in Fine Entity.

Fine:

Explanation: This ***weak entity*** is used to track fines imposed on the Members Entity. FineDate and Member ID will be composite in the relational schema as the primary key, ensuring each fine record has a unique date. FineAmount stores the fine amount, and PaymentStatus indicates whether the fine has been paid or not.

Resource:

Explanation: This entity describes information about library resources. ResourceID acts as the primary key, ensuring each resource has a unique ID. Book title stores the resource's title, ResourceType indicates the type of resource (e.g., book, DVD, video), ClassNumber records the resource's classification number, LoanPeriod specifies the borrowing period (in-library show 0 day), and FloorNumber and ShelfNumber are used to locate the resource within the library.
"/AvaliableQuantity" in Resources can be derived from Loan Entity.
"/TotalCopies" in Resource can be derived from ResourceCopies Entity.

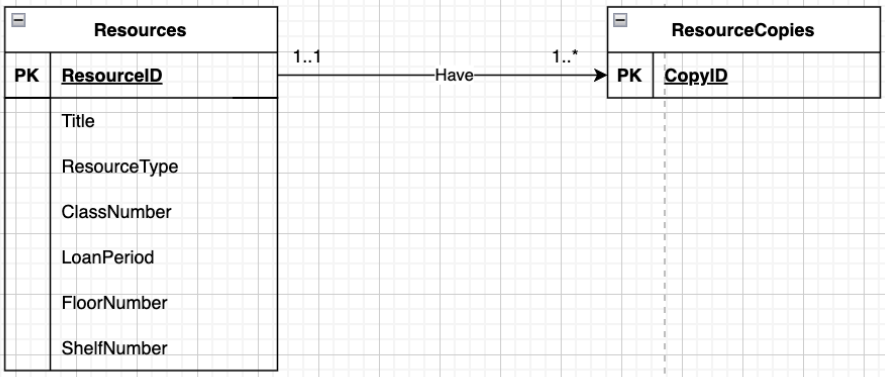
ResourceCopies:

Explanation: This ***weak entity*** is used to manage copies of resources, so it is based on Resource Entity. CopyID is a unique number for each book which combines ResourceID to ensure each resource copy has a unique ID.

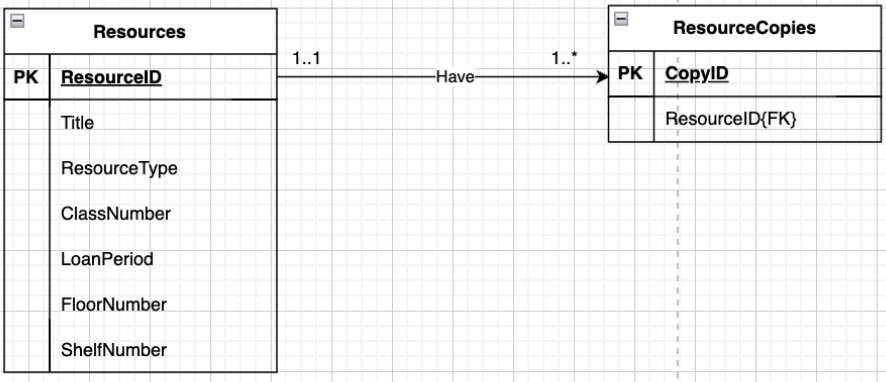
1b) Relational Database Schema

Step1: 1:N relationship between Resources and ResourceCopies

conceptual ERD



mapping process



Logical schema

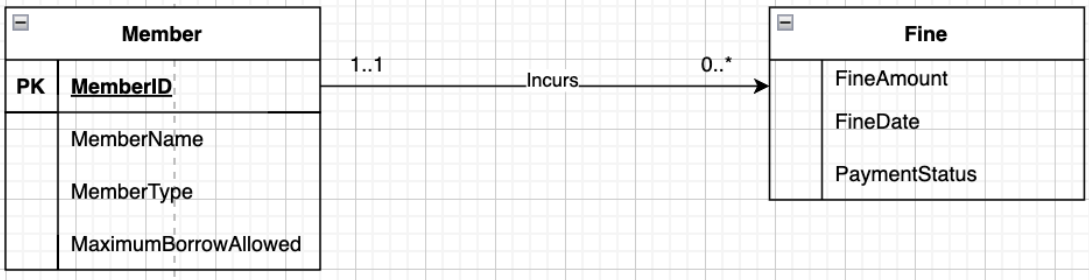
Resources(ResourceID{PK}, Title, ResourceType, ClassNumber, LoanPeriod, FloorNumber, ShelfNumber)
ResourceCopies(CopyID{PK}, ResourceID{FK})

Note:

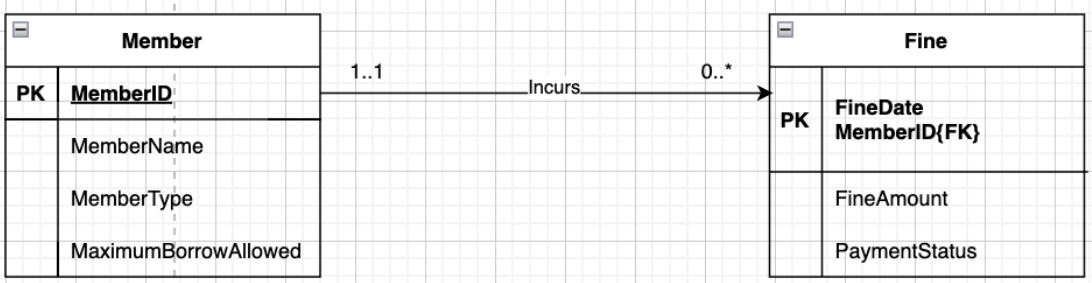
ResourceID is a foreign key to Resources.

Step2: 1:N relationship between Members and Fines

conceptual ERD



mapping process



Logical schema

Member(MemberID{PK}, MemberName, MemberType, MaximumBorrowAllowed)

Fine((MemberID{FK}, FineDate){PK}, FineAmount, PaymentStatus)

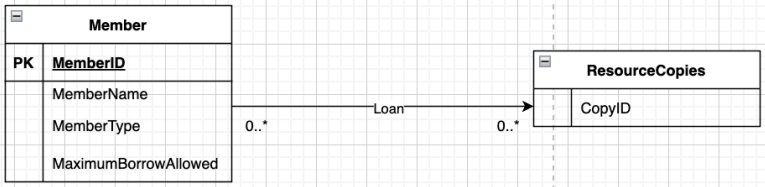
Note:

Fine is a weak entity, having a composite Primary Key.

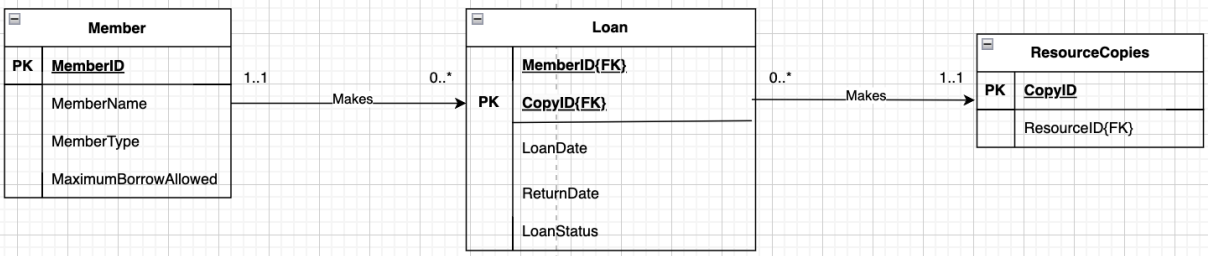
MemberID is a foreign key to Member.

Step3: N:M relationship, between Members and ResourceCopies

conceptual ERD



mapping process

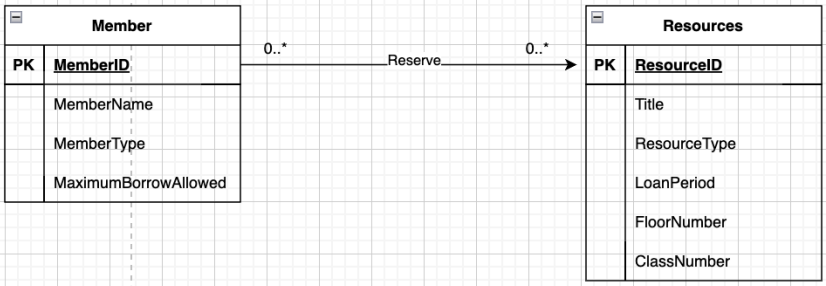


Logical schema

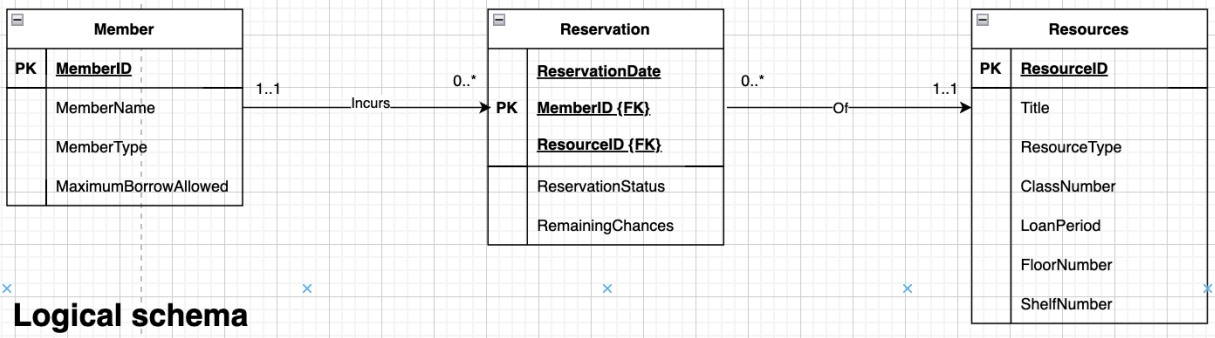
Member(MemberID{PK}, MemberName, MemberType, MaximumBorrowAllowed)
ResourceCopies(CopyID{PK}, ResourceID{FK})
Loan((ResourceID{FK}, CopyID{FK}, MemberID{FK}){PK}, LoanDate, ReturnDate, Status)
Note:
Created a new Loan entity for N:M relationship.
MemberID are CopyID are foreign keys, which represent Member and ResourceCopies.

Step4: N:M relationship, between Members and Resource

conceptual ERD



mapping process



Logical schema

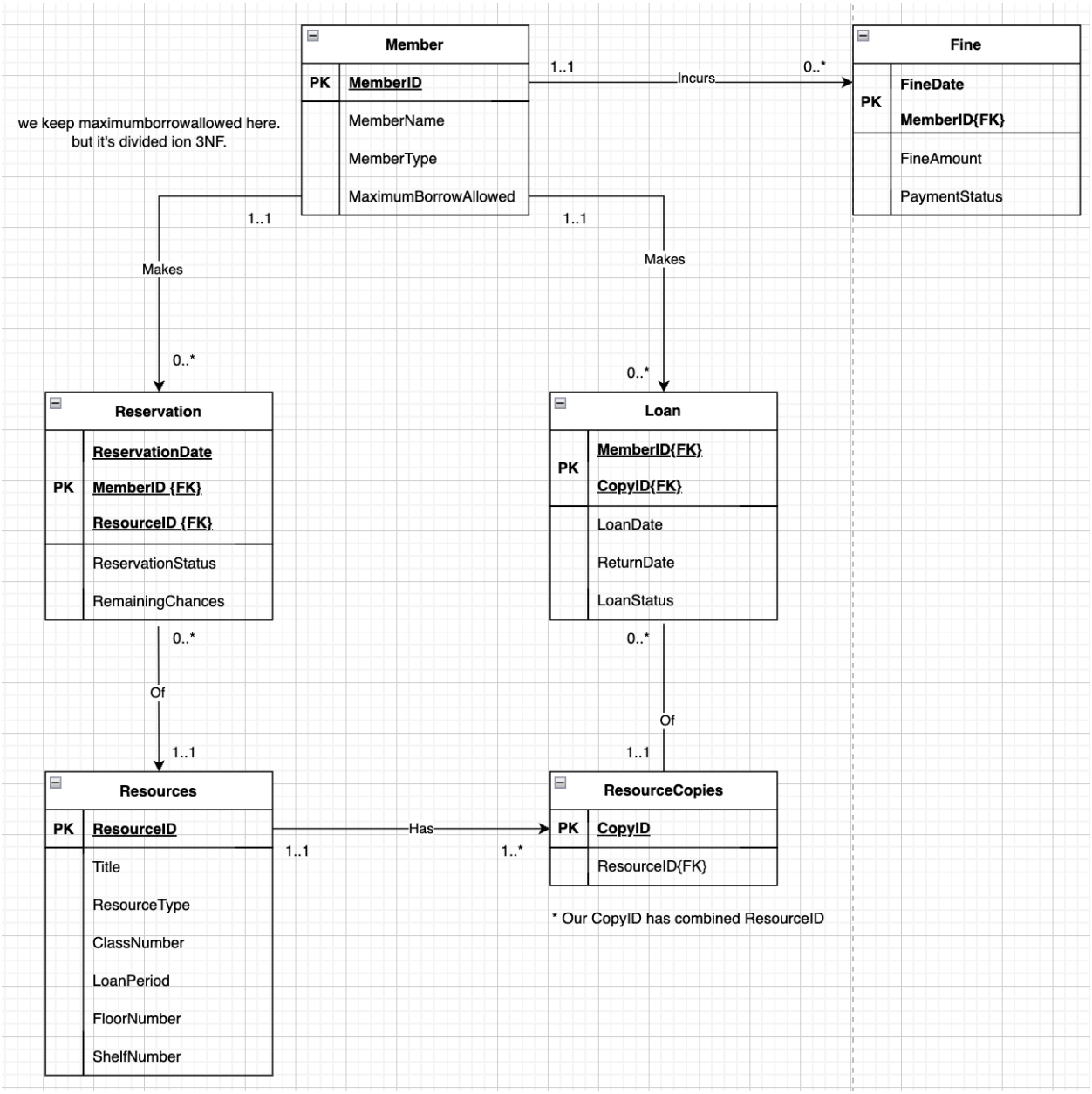
Member(MemberID{PK}, MemberName, MemberType, MaximumBorrowAllowed)

Resources(ResourceID{PK}, Title, ResourceType, ClassNumber, LoanPeriod, FloorNumber, ShelfNumber)

Reservation((MemberID{FK}, ResourceID{FK}, ReservationDate) {PK}, ReservationStatus, RemainingChances)

Note:
Created a new entity Reservation for N:M relationship.
MemberID and ResourceID are foreign keys, which represent Member and Resources.

Final Relational Schema:



*Red attributes are Primary Key

Fine ((FineDate, MemberID{FK}{PK}), FineAmount, PaymentStatus)

Reservation((ReservationDate, MemberID{FK}, ResourceID{FK}) {PK}, ReservationStatus, RemainingChances)

Loan(CopyID{FK}, MemberID{FK}){PK}, LoanDate, ReturnDate, LoanStatus)

Member(MemberID{PK},MemberName, MemberType, MaximumBorrowAllowed)

Resources(ResourceID{PK}, Title, ResourceType, ClassNumber, LoanPeriod, FloorNumber, ShelfNumber)
ResourceCopies(CopyID{PK}, ResourceID{FK})

1c) Normalised Design

- The normalised design for your application. You should produce a normalised design of your application, and explain why it is in 3rd Normal Form. To do this well, it is not sufficient simply to write out your relational schema, possibly with changes, and simply state the design is in 3rd Normal Form. Instead you should start the design again from the beginning using the Universal Relation (the set of all the data items required for the application), and apply the normalisation process described in the Normalisation lecture.

Please see the following 5 steps for our normalisation process.

1) Create A Universal Relation and Make Sure It Is 1NF

1NF:All attributes are atomic, no repeating groups.

UniversalRelation(MemberID, MemberName, MemberType, MaximumBorrowAllowed, ResourceID, BookTitle, ResourceType, ClassNumber, LoanPeriod, Location(FloorNumber, ShelfNumber), CopyID, ReservationDate, ReservationStatus, LoanDate, ReturnDate, LoanStatus, FineDate, FineAmount, PaymentStatus)

- **Location(FloorNumber, ShelfNumber) is not atomic, so we divide it into two attributes.**

UniversalRelation(MemberID, MemberName, MemberType, MaximumBorrowAllowed, ResourceID, BookTitle, ResourceType, ClassNumber, LoanPeriod, FloorNumber, ShelfNumber, CopyID, ReservationDate, ReservationStatus, LoanDate, ReturnDate, LoanStatus, FineDate, FineAmount, PaymentStatus)

2) Identify The Primary Key of The Universal Relation

We identify MemberID, ResourceID, CopyID as primary keys.
The primary key of this relation is the combination of MemberID, ResourceID, CopyID.

3) Create a Set of 2NF Relations By Separating All Attributes That Are Only Partially Dependent On the Primary Key of The Universal Relation into Separate

(2NF:No partial dependencies. All non-key attributes are fully functionally dependent on the primary key.)

Attributes dependent on primary keys:

- **Member**(MemberID {PK}, MemberName, MemberType, MaximumBorrowAllowed, , FineDate, FineAmount, PaymentStatus)

- **ResourceCopies**(CopyID, ResourceID{PK}, BookTitle, ResourceType, ClassNumber, LoanPeriod, FloorNumber, ShelfNumber,)

Attributes dependent on composite primary keys:
We also identified ReservationDate and FineDate as partial keys, adding them with primary keys to compose composite primary keys.

- **Loan**((CopyID, MemberID){PK}, ReservationDate, ReservationStatus, LoanDate, ReturnDate, LoanStatus)

- **Reservation**((MemberID, ResourceID, ReservtionDate) {PK}, ReservationStatus)

- **Fine**((MemberID, FineDate){PK}, FineAmount, PaymentStatus)

4) Create A Set of 3NF Relations By Separating Out Any Attributes That Are Transitively Dependent on Primary Keys In The Set of 2NF Relations

a) From 2NF:

- **ResourceCopies**(CopyID, ResourceID{PK}, BookTitle, ResourceType, ClassNumber, LoanPeriod, FloorNumber, ShelfNumber,)

To 3NF:

- **ResourceCopies**(CopyID{PK}, ResourceID)
- **Resources** (ResourceID{PK}, BookTitle, ResourceType, ClassNumber, LoanPeriod, FloorNumber, ShelfNumber) with Primary Key ResourceID.

b) From 2NF:

- **Member**(MemberID {PK}, MemberName, MemberType, MaximumBorrowAllowed, , FineDate, FineAmount, PaymentStatus)

To 3NF:

- **Member**(MemberID {PK}, MemberName, MemberType, FineDate, FineAmount, PaymentStatus)
- **Membertype**(MemberType{PK}, MaximumBorrowAllowed)

5) The Final Design with A Set of 3NF Relations:

- **Member**(MemberID{PK},MemberName, MemberType)
- **Membertype**(MemberType{PK}, MaximumBorrowAllowed)
- **Loan**((CopyID, MemberID){PK}, LoanDate, ReturnDate, LoanStatus)
- **Reservation**((MemberID, ResourceID, ReservtionDate) {PK}, ReservationStatus)
- **Fine**(MemberID, FineDate{PK}, FineAmount, PaymentStatus)
- **ResourceCopies**(CopyID{PK},ResourceID)
- **Resources**(ResourceID{PK}, BookTitle, ResourceType, ClassNumber, LoanPeriod, FloorNumber, ShelfNumber)

Part 2: DATABASE IMPLEMENTATION

2a) 'Create table' Command

– We created 8 tables and implemented declarative constraints

– The following code is also included in Create_table.sql

```
CREATE TABLE Member (
MemberID INT PRIMARY KEY,
MemberName VARCHAR2(50) NOT NULL,
MemberType VARCHAR2(50) NOT NULL,
CONSTRAINT MemberTypeCheck CHECK(MemberType IN ('Student',
'Staff'))
);

CREATE TABLE MemberType (
MemberType VARCHAR2(50) PRIMARY KEY,
MaximumBorrowAllowed INT NOT NULL,
CONSTRAINT MemberType_Check CHECK(MemberType IN ('Student',
'Staff')),
CONSTRAINT MaximumBorrowAllowedCheck
CHECK((MemberType = 'Student' AND MaximumBorrowAllowed = 5) OR
(MemberType = 'Staff' AND MaximumBorrowAllowed = 10))
);

CREATE TABLE Resources (
ResourceID INT PRIMARY KEY,
Title VARCHAR2(50) NOT NULL,
ResourceType VARCHAR2(25) NOT NULL,
ClassNumber INT NOT NULL,
LoanPeriod INT NOT NULL,
FloorNumber INT NOT NULL,
ShelfNumber VARCHAR2(50) NOT NULL,
CONSTRAINT LoanPeriodcheck CHECK(LoanPeriod >= 0 and LoanPeriod
<=14),
CONSTRAINT FloorNumbercheck CHECK(FloorNumber >= 1 and
FloorNumber <=3)
);

CREATE TABLE ResourceCopies (
```

```
CopyID INT PRIMARY KEY,
ResourceID INT NOT NULL,
CONSTRAINT FK_ResourceID FOREIGN KEY (ResourceID) REFERENCES
Resources(ResourceID)
);

CREATE TABLE Fine (
MemberID INT NOT NULL,
FineDate DATE NOT NULL,
FineAmount INT DEFAULT 1,
PaymentStatus VARCHAR2(50) NOT NULL,
PRIMARY KEY(MemberID, FineDate),
CONSTRAINT FK_MemberID FOREIGN KEY (MemberID) REFERENCES
Member(MemberID)
);

CREATE TABLE Reservation (
MemberID INT NOT NULL,
ResourceID INT NOT NULL,
ReservationDate DATE NOT NULL,
ReservationStatus VARCHAR2(20) NOT NULL,
RemainingChances INT NOT NULL,
PRIMARY KEY (MemberID, ResourceID, ReservationDate),
CONSTRAINT FK_MemberID_Reservation FOREIGN KEY (MemberID)
REFERENCES Member(MemberID),
CONSTRAINT FK_ResourceID_Reservation FOREIGN KEY (ResourceID)
REFERENCES Resources(ResourceID),
CONSTRAINT RemainingChancescheck CHECK(RemainingChances >= 0 and
RemainingChances <=3)
);


CREATE TABLE Loan (
MemberID INT NOT NULL,
CopyID INT NOT NULL,
LoanDate DATE NOT NULL,
ReturnDate DATE,
LoanStatus VARCHAR2(25) NOT NULL,
PRIMARY KEY (MemberID, CopyID,LoanDate),
CONSTRAINT FK_MemberID_Loan FOREIGN KEY (MemberID) REFERENCES
Member(MemberID),
```

```
CONSTRAINT FK_CopyID_Loan FOREIGN KEY (CopyID) REFERENCES
ResourceCopies(CopyID)
);


CREATE TABLE LoanHistory (
MemberID INT NOT NULL,
CopyID INT NOT NULL,
LoanDate DATE NOT NULL,
ReturnDate DATE,
PRIMARY KEY (MemberID, CopyID,LoanDate)
);
```

2b) Sample Test Data

- Please check the attached Insert_data.sql for our test data.

- The following are screenshots for reference.

1. Member

MEMBERID	MEMBERNAME	MEMBERTYPE
201	John Doe	Student
202	Jane Smith	Staff
203	Bob Johnson	Student
204	Alice Brown	Staff
205	Charlie Davis	Student
206	Eva White	Staff
207	David Lee	Student
208	Grace Miller	Staff
209	Frank Wilson	Student
210	Helen Taylor	Staff

2. MemberType

MEMBERTYPE	MAXIMUMBORROWALLOWED
Student	5
Staff	10

3. Resources

RESOURCEID	TITLE	RESOURCETYPE	CLASSNUMBER	LOANPERIOD	FLOORNUMBER	SHELFNUMBER
1	Introduction to Computer Science	Book	1	14	2	A200
2	The Art of Programming	Book	1	14	2	B210
3	Chemistry Basics	Book	2	14	3	C300
4	History of Ancient Civilizations	DVD	3	14	1	A100
5	Physics for Beginners	Book	2	14	3	B310
6	Web Development Guide	Book	1	14	2	B220
7	Introduction to Poetry	DVD	4	2	1	B110
8	Environmental Science Explained	Video	7	2	2	B230
9	Theatre and Drama Techniques	CD	8	14	1	B130
10	Language Learning Essentials	Book	9	0	3	B340

4. ResourceCopies

COPYID	RESOURCEID
1001	1
1002	1
2001	2
3001	3
3002	3
4001	4
5001	5
5002	5
5003	5
6001	6
7001	7
8001	8
9001	9
10001	10
10002	10

5. Fine

MEMBERID	FINEDATE	FINEAMOUNT	PAYMENTSTATUS
204	04-OCT-23	2	Paid
205	27-NOV-23	1	Paid
209	21-NOV-23	1	Unpaid
209	22-NOV-23	1	Unpaid
209	23-NOV-23	1	Unpaid
209	24-NOV-23	1	Unpaid
209	25-NOV-23	1	Unpaid
209	26-NOV-23	1	Unpaid
209	27-NOV-23	1	Unpaid
209	28-NOV-23	1	Unpaid
209	29-NOV-23	1	Unpaid
209	30-NOV-23	1	Unpaid
210	30-NOV-23	1	Unpaid

6. Loan

MEMBERID	COPYID	LOANDATE	RETURNDATE	LOANSTATUS
205	8001	24-NOV-23	27-NOV-23	Returned
209	6001	06-NOV-23	-	Overdue
210	2001	15-NOV-23	-	Overdue
204	5001	20-NOV-23	-	Checkedout
201	5002	23-NOV-23	-	Checkedout
205	3001	25-NOV-23	-	Checkedout
205	1001	28-NOV-23	-	Checkedout
207	4001	20-NOV-23	-	Checkedout
203	7001	24-NOV-23	25-NOV-23	Returned
208	7001	26-NOV-23	27-NOV-23	Returned
206	8001	27-NOV-23	29-NOV-23	Returned

7. Reservation

MEMBERID	RESOURCEID	RESERVATIONDATE	RESERVATIONSTATUS	REMAININGCHANCES
202	7	02-OCT-23	Completed	3
206	4	06-OCT-23	Completed	3
201	9	14-OCT-23	Completed	3
204	3	08-OCT-23	Completed	3
208	7	24-NOV-23	Completed	3
206	8	25-NOV-23	Completed	3
210	6	12-NOV-23	Waitlisted	3
204	2	16-NOV-23	Waitlisted	3
210	1	29-NOV-23	Waitlisted	3
203	4	12-OCT-23	Completed	3
205	6	11-OCT-23	Cancelled	0
205	7	05-OCT-23	Cancelled	0
203	8	28-NOV-23	Notified	3
202	7	26-NOV-23	Notified	2

8. LoanHistory

MEMBERID	COPYID	LOANDATE	RETURNDATE
204	7001	01-OCT-23	04-OCT-23
202	7001	04-OCT-23	06-OCT-23
209	4001	04-OCT-23	10-OCT-23
206	4001	10-OCT-23	18-OCT-23
205	9001	12-OCT-23	16-OCT-23
201	9001	16-OCT-23	19-OCT-23
210	3001	04-OCT-23	18-OCT-23
202	3002	06-OCT-23	15-OCT-23
204	3002	15-OCT-23	20-OCT-23
207	4001	10-OCT-23	22-OCT-23
203	4001	22-OCT-23	02-NOV-23
204	8001	01-OCT-23	04-OCT-23
201	6001	02-OCT-23	12-OCT-23

2c) View Definitions

- We establish three user-specific views with clear lists of outputs

- Please check the attached View.sql for our SQL code for views

- The following are screenshots for reference.

1. MemberDetails View
    - This view merges details from the ‘Member’ and ‘Fine’ tables, offering a comprehensive overview of each member. The ‘Member status’ is ‘Suspend’ if total fine >=10

MEMBERID	MEMBERNAME	MEMBERTYPE	TOTALFINE	MEMBERSTATUS
210	Helen Taylor	Staff	1	Active
207	David Lee	Student	0	Active
205	Charlie Davis	Student	0	Active
204	Alice Brown	Staff	0	Active
203	Bob Johnson	Student	0	Active
208	Grace Miller	Staff	0	Active
202	Jane Smith	Staff	0	Active
206	Eva White	Staff	0	Active
209	Frank Wilson	Student	10	Suspend
201	John Doe	Student	0	Active

2. ResourceStatus View
    - This view provides information about each resource, including the number of copies and available current available copies.

RESOURCEID	TITLE	TOTALCOPIES	REMAININGCOPIES
1	Introduction to Computer Science	2	1
2	The Art of Programming	1	0
3	Chemistry Basics	2	1
4	History of Ancient Civilizations	1	0
5	Physics for Beginners	3	1
6	Web Development Guide	1	0
7	Introduction to Poetry	1	1
8	Environmental Science Explained	1	1
9	Theatre and Drama Techniques	1	1
10	Language Learning Essentials	2	2

3. FineSummary View:

- This view provides a summary of fines, including member details.

MEMBERID	MEMBERNAME	TOTALFINEAMOUNT	LASTFINEDATE
204	Alice Brown	2	04-OCT-23
209	Frank Wilson	10	30-NOV-23
205	Charlie Davis	1	27-NOV-23
210	Helen Taylor	1	30-NOV-23

2d) SQL Queries with Output Listings

- We created a set of 12 user-specific SQL queries demonstrating a wide range of SQL language constructs with clear lists of outputs
- Please check the attached Query.sql for our SQL code for Queries
- The following are screenshots for reference.

1. List of Members with Overdue Loans:

MEMBERID	MEMBERNAME	COPYID	LOANSTATUS
209	Frank Wilson	6001	Overdue
210	Helen Taylor	2001	Overdue

2. List of Top 5 resources that are often reserved:

RESOURCEID	TITLE	RESERVATIONCOUNT
7	Introduction to Poetry	4
6	Web Development Guide	2
4	History of Ancient Civilizations	2
8	Environmental Science Explained	2
1	Introduction to Computer Science	1

3. List of Members with Unpaid Fines:

MEMBERID	MEMBERNAME	TOTALUNPAIDFINES
209	Frank Wilson	10
210	Helen Taylor	1

4. The Top 5 most popular resources:

- This query identifies the total number of loans (including Loan and LoanHistory) for Resources

RESOURCEID	TITLE	TOTALLOANCOUNT
4	History of Ancient Civilizations	5
7	Introduction to Poetry	4
3	Chemistry Basics	4
8	Environmental Science Explained	3
6	Web Development Guide	2

5. The Top 5 Members who often borrow resources from the library:

- This query identifies the total number of loans (including Loan and LoanHistory) for members

MEMBERID	MEMBERNAME	TOTALLOANS
204	Alice Brown	4
205	Charlie Davis	4
201	John Doe	3
210	Helen Taylor	2
209	Frank Wilson	2

6. List of Resources with No Reservations:

RESOURCEID	TITLE	RESOURCETYPE
5	Physics for Beginners	Book
10	Language Learning Essentials	Book

7. List of Resources with Multiple Copies:

- This query identifies the loan situation of resources with more than one copy

RESOURCEID	TITLE	COPIESNO	CHECKOUTNO
1	Introduction to Computer Science	2	1
3	Chemistry Basics	2	1
5	Physics for Beginners	3	2
10	Language Learning Essentials	2	0

8. Total Number of Loans per Floor:

- This query provides the total number of loans for each floor in the library

FLOORNUMBER	TOTALLOANS
1	1
2	3
3	3

9. Cancellation of Reservation:

- This query identifies the member who cancels the reservation (fails to take up) the most

MEMBERID	MEMBERNAME	TOTALCANCELLATION
205	Charlie Davis	2

10. Remaining BorrowAmount:

- This query calculates the remaining amount of resources each member can borrow

MEMBERID	MEMBERNAME	REMAININGQUOTA
201	John Doe	4
202	Jane Smith	10
203	Bob Johnson	5
204	Alice Brown	9
205	Charlie Davis	3
206	Eva White	10
207	David Lee	4
208	Grace Miller	10
209	Frank Wilson	4
210	Helen Taylor	9

11. Take up the resources from the Reservation:

- This query identifies the member who has successfully accessed the resource by making reservations from the library the most

MEMBERID	MEMBERNAME	SUCCESSFULRESERVATIONS
206	Eva White	2
204	Alice Brown	1
208	Grace Miller	1
201	John Doe	1
202	Jane Smith	1

12. List of resources with Most Overdue Loans:

- This query identifies resources with the highest number of overdue loans

RESOURCEID	TITLE	NUMOVERDUELOANS
2	The Art of Programming	1
6	Web Development Guide	1
4	History of Ancient Civilizations	0
9	Theatre and Drama Techniques	0
1	Introduction to Computer Science	0