ADDIS ABABA UNIVERSITY
አዲስ፡አበባ፡ዩኒቨርሲቲ
SEEK WISDOM,ELEVATE YOUR INTELLECT AND SERVE HUMANITY !

ADDIS ABABA INSTITUTE OF TECHNOLOGY

DATABASE GROUP ASSIGNMENT SECTION 3

GROUP MEMBERS	**ID NUMBER**
1. HAILEMARYAM DAGNEW	UGR/0344/16
2. HEARMON TESFAY	UGR/4747/16
3. OUMER JEMAL	UGR/4858/16
4. AMANUEL DEJENE	UGR/9904/16
5. HIRUY LEGESSE	UGR/1838/16

SUBMITTED TO: Mr. Segni

SUBMISSION DATE: 12 JUNE 2025 G.C

1.Introduction

1.1 Project Background

In today’s digital age, efficient data management is crucial for organizations to streamline operations and enhance service delivery. Libraries handle vast amounts of data, including member records, book inventories, loans, and employee information. Managing this data manually is time-consuming, error-prone, and inefficient.

1.2 Purpose

This project focuses on designing a **Library Management System (LMS) database** for *Selam Library*, a community library facing challenges with its current paper-based record-keeping system. The proposed database will automate key processes such as book tracking, loan management, member registration, and reporting, ensuring accuracy, speed, and reliability.

1.3 Objectives

1. **Replace manual processes** with a structured digital database.
2. **Improve data accuracy** by reducing human errors in book loans and returns.
3. **Enhance accessibility** of members and book records for staff.
4. **Generate real-time reports** on overdue books, popular titles, and member activity.
5. **Optimize inventory management** by tracking book conditions and availability.

By implementing this database, Selam Library will modernize its operations, improve user experience, and maintain better control over its resources. The following sections outline the database design, from conceptualization to implementation, including entity relationships, SQL table structures, and sample queries.

1.4 Key Features of the Database

- **Member Management:** Store and retrieve member details efficiently.
- **Book Cataloging:** Maintain an up-to-date record of all books and their copies.
- **Loan Tracking:** Monitor due dates, returns, and fines automatically.
- **Employee Records:** Manage staff information and roles.
- **Reporting Tools:** Generate insights into library operations.

This project demonstrates how a well-structured database can transform library management, ensuring smooth and error-free operations.

2. Background

2.1 Company Profile

Selam Library is a community-focused public library located in the heart of the city, serving residents of all ages since 2005. With a collection of over 50,000 books across various genres and subjects, Selam Library has established itself as a center for learning, research, and community engagement.

2.2 Mission

To provide free and equal access to information, foster lifelong learning, and serve as a cultural hub for our diverse community.

2.3 Vision:

To be the leading public library in the region by offering innovative services, maintaining an up-to-date collection, and creating an inclusive environment that meets the evolving needs of our patrons.

The library currently employs 15 staff members including librarians, assistants, and administrative personnel. It serves approximately 5,000 registered members with an average of 300 daily visitors. The current manual record-keeping system has become inefficient as the library's operations have grown in scale and complexity.

3. Requirement Analysis

3.1 Current Problems:

1. **Inefficient Loan Tracking:** The manual ledger system for tracking book loans leads to frequent errors, lost records, and difficulties in monitoring overdue items.
2. **Limited Member Management:** Maintaining member information in paper files makes it hard to track membership status, contact details, and borrowing history.
3. **Inventory Challenges:** Without a proper cataloging system, staff struggle to locate books, track available copies, and monitor book conditions.
4. **Reporting Difficulties:** Generating reports on popular books, overdue items, or member activity requires time-consuming manual compilation.
5. **Staff Management:** Employee records, including positions and salaries, are maintained separately from operational data, causing administrative inefficiencies.

3.2 Proposed Database Solution Functions:

1. **Automated Loan Management:** Track all book loans including due dates, returns, and fines with automatic calculations and notifications.
2. **Comprehensive Member Database:** Maintain up-to-date member information with borrowing history and contact details.
3. **Inventory Control:** Manage the complete book collection including multiple copies of titles, their conditions, and availability status.
4. **Staff Management System:** Centralize employee records with their roles and supervision hierarchy.
5. **Reporting Module:** Generate various reports including:
   - Overdue books and fines
   - Most popular titles
   - Member activity
   - Inventory status
   - Staff performance metrics
6. **Search Functionality:** Enable quick search of books by title, author, ISBN, or category.
7. **Automated Notifications:** System alerts for due dates, overdue items, and membership renewals.

3.3 Technical Requirements

The database must adhere to normalization principles (1NF, 2NF, 3NF) to eliminate redundancy, ensure data integrity, and optimize performance. This includes:

- Storing atomic values (1NF).
- Removing partial dependencies (2NF).
- Eliminating transitive dependencies (3NF).

Pros:

- Sets clear expectations early.
- Justifies the database structure in later sections.

The new database system will replace the current paper-based operations, reducing errors, saving time, and improving service quality for both library staff and patrons. It will provide real-time information about book availability, streamline the borrowing process, and enhance overall library management.

4. Conceptual Design

This design outlines the structure and relationships between different entities to ensure efficient data storage and retrieval

Entities and Attributes

1. Employee

Description:

Represents library staff members and their organizational hierarchy. Attributes:

- **EmployeeID** (PK) – Unique identifier for each employee
- **Fname** – Employee's first name
- **Mname** – Employee's middle name (optional)
- **Lname** – Employee's last name
- **Position** – Job role (e.g., Librarian, Assistant, Manager)
- **Salary** – Monthly salary
- **SuperEmployeeID** (FK → Employee) – Manager/Supervisor reference

2. Member

Description:

Stores information about library members eligible to borrow books. Attributes:

- **MemberID** (PK) – Unique identifier for each library member
- **Fname** – Member's first name
- **Lname** – Member's last name
- **Address** – Member's residential address
- **MembershipDate** – Date when the member joined

3. Book

Description:

Store information about the books available in the library Attributes:

- **ISBN** (PK) – International Standard Book Number
- **Title** – Book title
- **Author** – Book author
- **Publisher** – Publishing company
- **Category** – Genre (e.g., Fiction, Science, History)
- **Year** – Publication year
- **TotalCopies** – Total copies owned by the library
- **AvailableCopies** – Copies currently available for loan

4. Book Copy(Weak Entity)

Description:

Tracks individual physical copies of books in the library's inventory Attributes:

- **CopyID** (PK) – Unique identifier for each physical copy
- **AcquisitionDate** – When the copy was added to the library
- **Condition** – Physical state (e.g., New, Good, Damaged)
- **B_ISBN** (FK → Book) – References the book's ISBN

5. Loan

Description:

Manages book borrowing transactions between members and the library. Attributes:

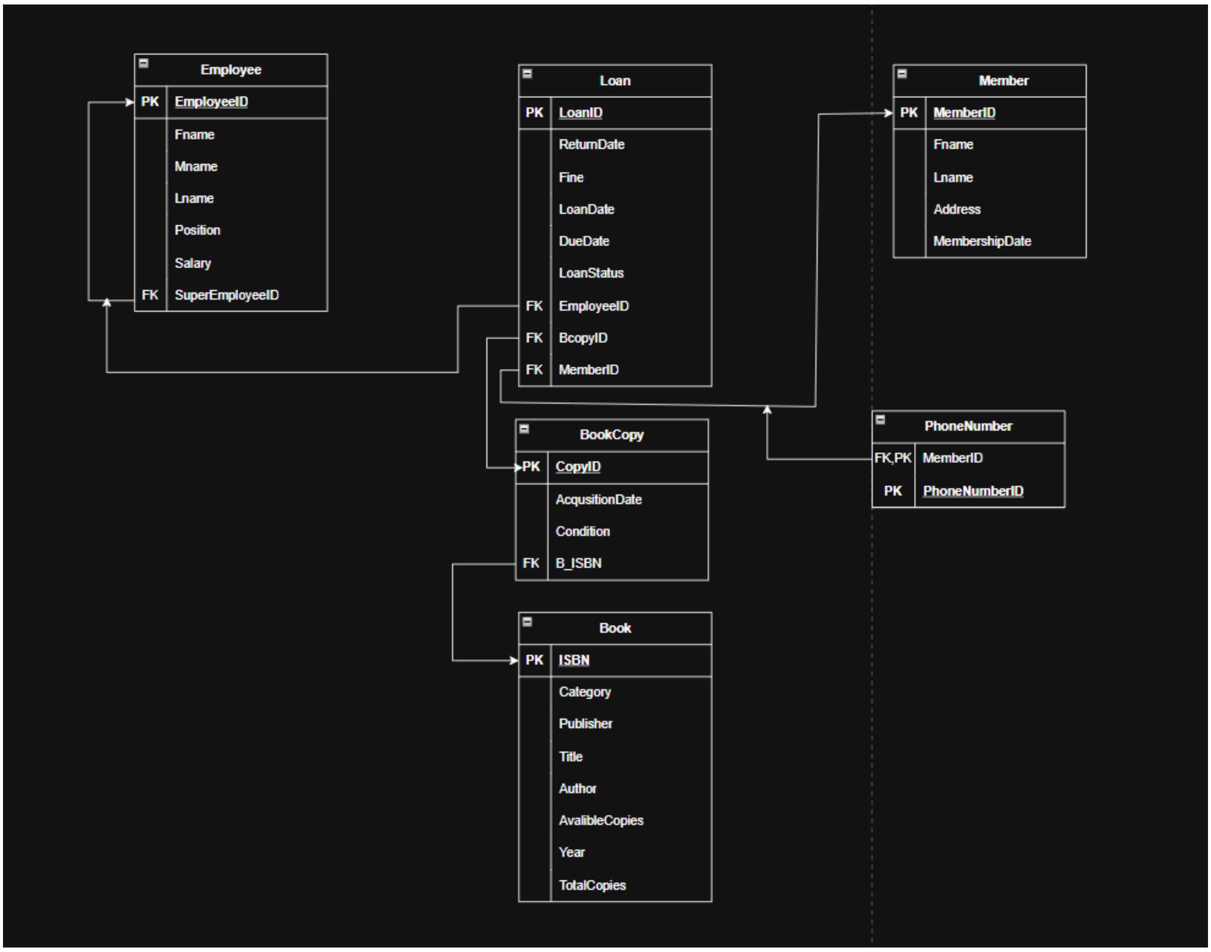
- **LoanID** (PK) – Unique loan transaction ID
- **LoanDate** – Date the book was borrowed
- **DueDate** – Expected return date
- **ReturnDate** – Actual return date (nullable if not returned)
- **Fine** – Calculated fine if overdue
- **LoanStatus** (e.g., Active, Returned, Overdue)
- **EmployeeID** (FK → Employee) – Who processed the loan
- **MemberID** (FK → Member) – Who borrowed the book
- **CopyID** (FK → BookCopy) – Which copy was loaned

6. PhoneNumber (Multivalued Attribute → Separate Entity)

Description:

Stores multiple contact numbers for members. Attributes:

- **PhoneID** (PK) – Unique phone number ID
- **PhoneNumber** – Contact number
- **MemberID** (FK → Member) – Links to the member



5. Relationships (PK & FK Rules)

Relationship	Primary Key (PK)	Foreign Key (FK)	Description
Employee → Employee	EmployeeID	SuperEmployeD	Self-referential (manager-subordinate)
Loan → Employee	LoanID	EmployeeID	Which employee processed the loan
Loan → Member	LoanID	MemberID	Which member borrowed the book

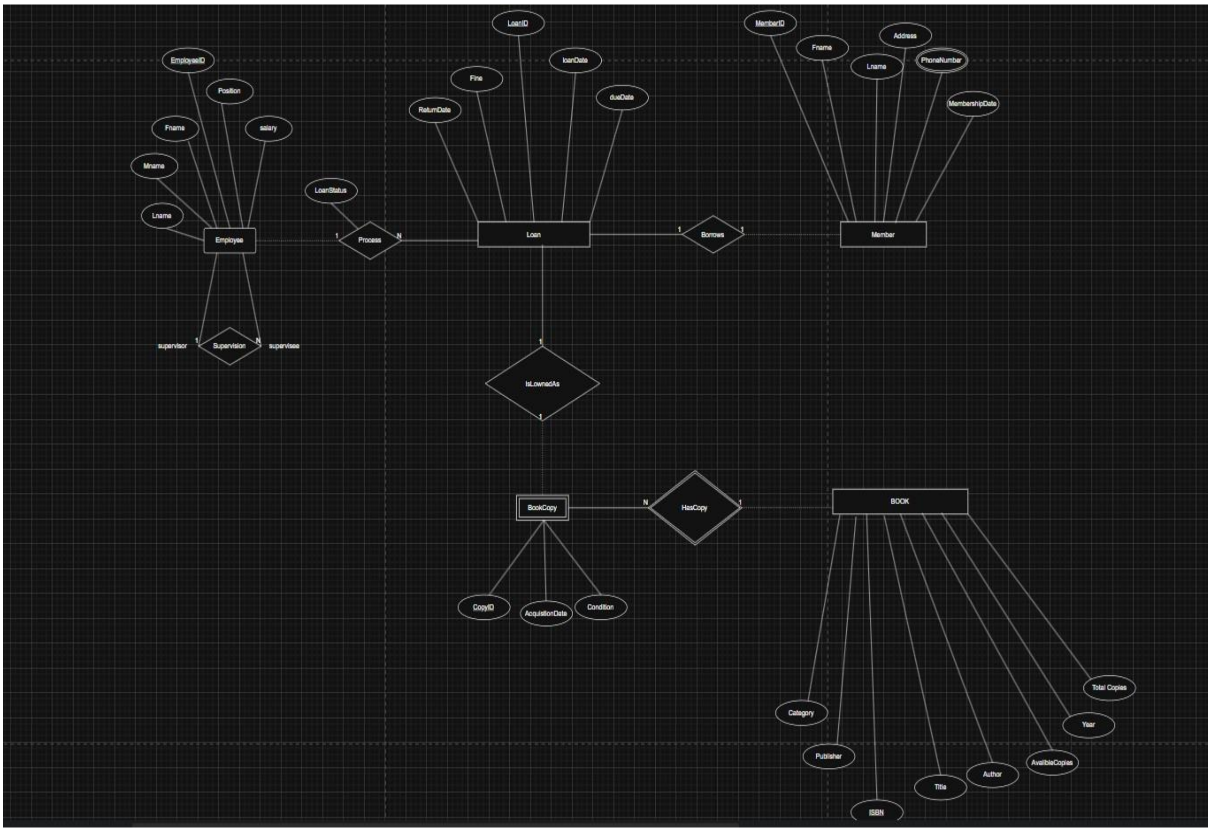
Loan → BookCopy	LoanID	CopyID	Which copy was loaned
BookCopy → Book	CopyID	B_ISBN	Which book the copy belongs to
PhoneNumber → Member	PhoneID	MemberID	Member’s contact number

6. Relationship Constraints

Relationship	**Cardinality**	**Participation**
Employee supervises Employee	One-to-Many (1:N)	Partial (not all employees supervise others)
Employee processes Loan	One-to-Many (1:N)	Total (every loan is processed by an employee)
Member borrows Loan	One-to-Many (1:N)	Partial (not all members have active loans)
Book has BookCopy	One-to-Many (1:N)	Total (every copy belongs to a book)
Loan involves BookCopy	One-to-One (1:1)	Total (every loan must have a copy)

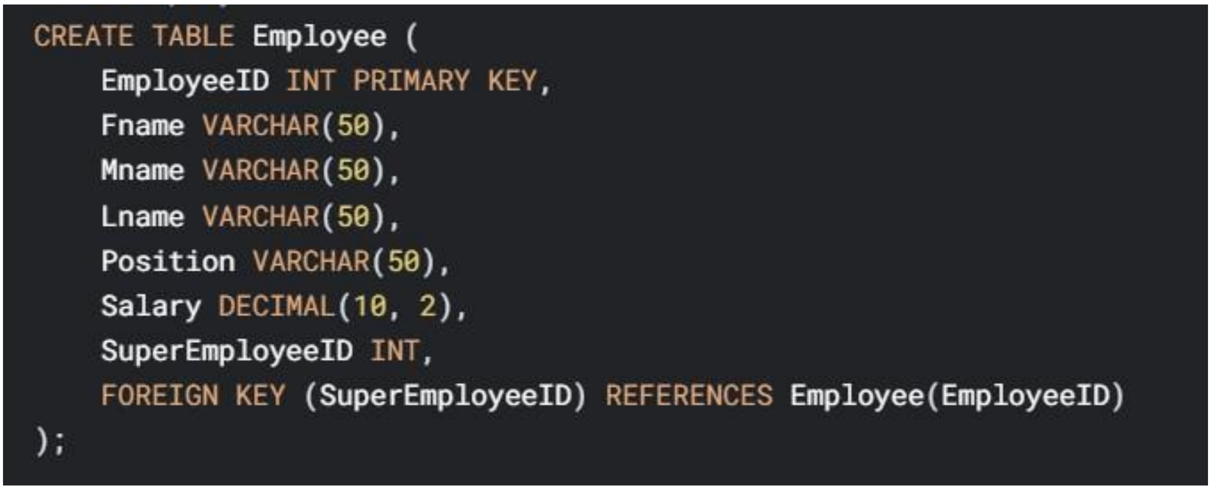
Member has PhoneNumber	One-to-Many (1:N)	Partial (some members may not provide a number)

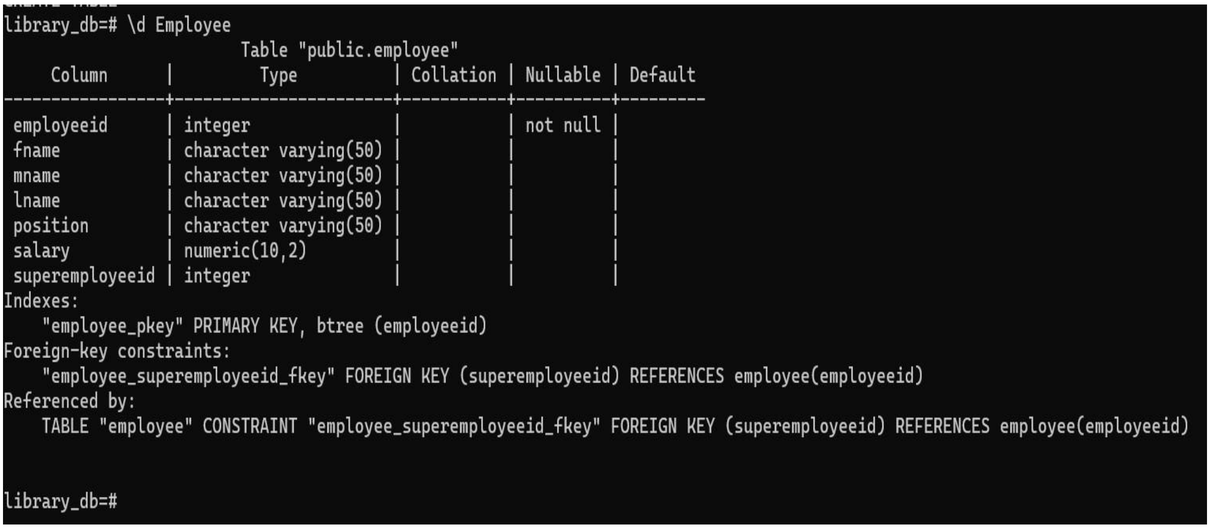
7. ER Diagram



8. Relational Tables

1. Employee Table

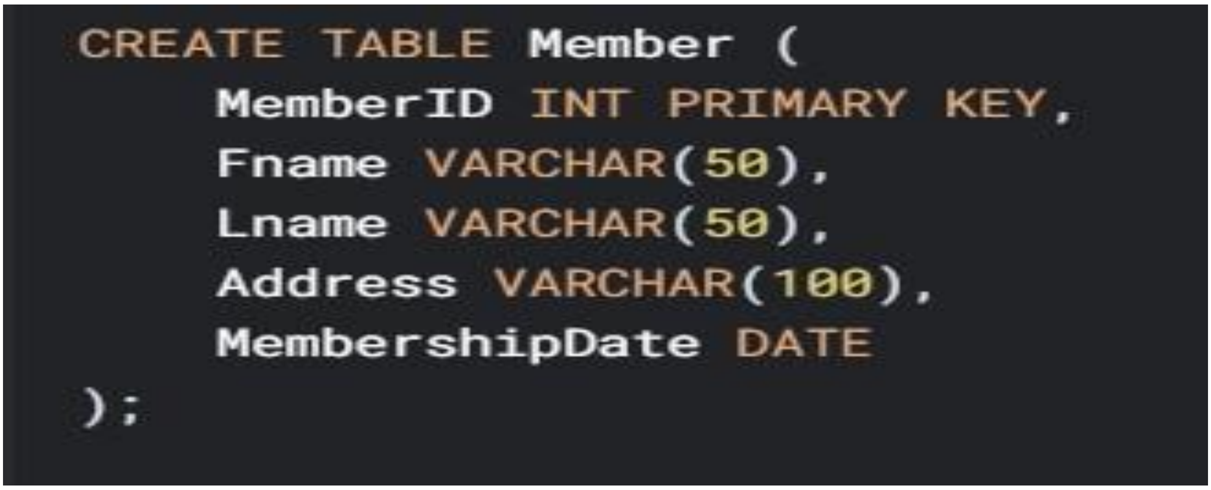

```
CREATE TABLE Employee (
    EmployeeID INT PRIMARY KEY,
    Fname VARCHAR(50),
    Mname VARCHAR(50),
    Lname VARCHAR(50),
    Position VARCHAR(50),
    Salary DECIMAL(10, 2),
    SuperEmployeeID INT,
    FOREIGN KEY (SuperEmployeeID) REFERENCES Employee(EmployeeID)
);
```



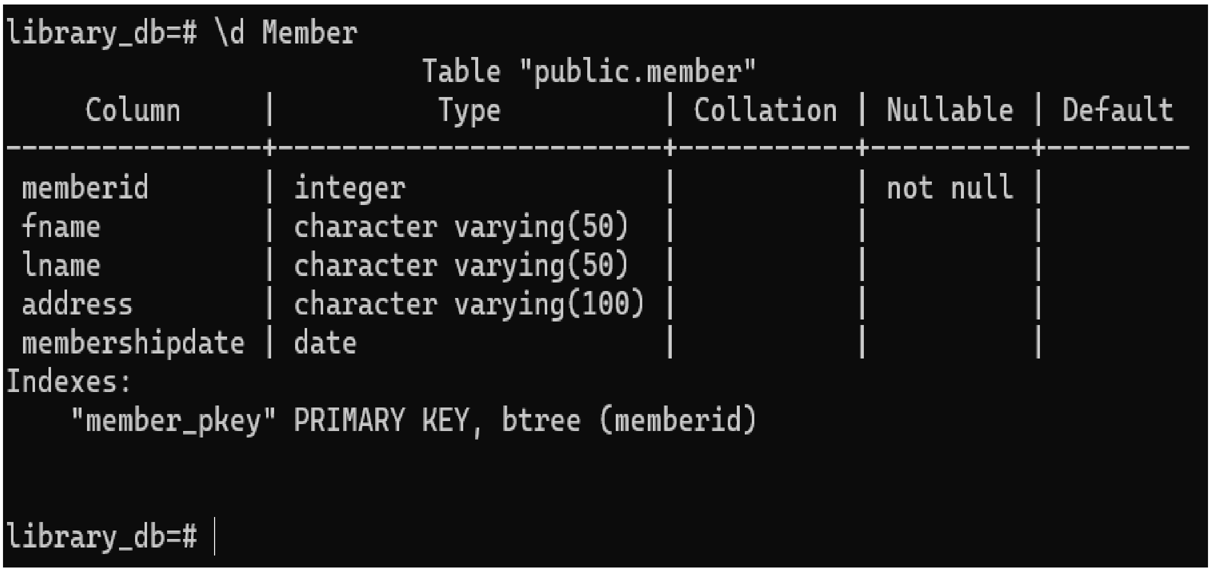
```
library_db=# \d Employee
                        Table "public.employee"
     Column     |         Type          | Collation | Nullable | Default
----------------+-----------------------+-----------+----------+---------
 employeeid     | integer               |           | not null |
 fname          | character varying(50) |           |          |
 mname          | character varying(50) |           |          |
 lname          | character varying(50) |           |          |
 position       | character varying(50) |           |          |
 salary         | numeric(10,2)         |           |          |
 superemployeeid | integer              |           |          |
Indexes:
    "employee_pkey" PRIMARY KEY, btree (employeeid)
Foreign-key constraints:
    "employee_superemployeeid_fkey" FOREIGN KEY (superemployeeid) REFERENCES employee(employeeid)
Referenced by:
    TABLE "employee" CONSTRAINT "employee_superemployeeid_fkey" FOREIGN KEY (superemployeeid) REFERENCES employee(employeeid)


library_db=#
```

2. Member Table



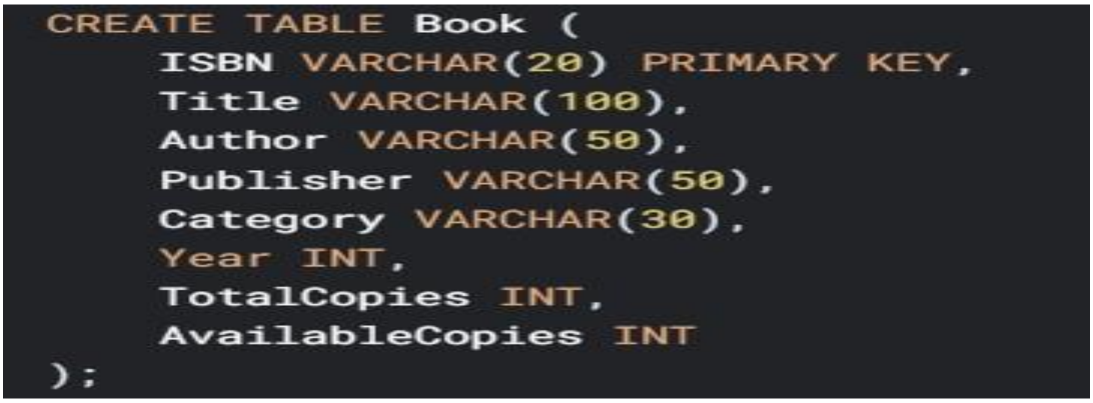
```
CREATE TABLE Member (
    MemberID INT PRIMARY KEY,
    Fname VARCHAR(50),
    Lname VARCHAR(50),
    Address VARCHAR(100),
    MembershipDate DATE
);
```



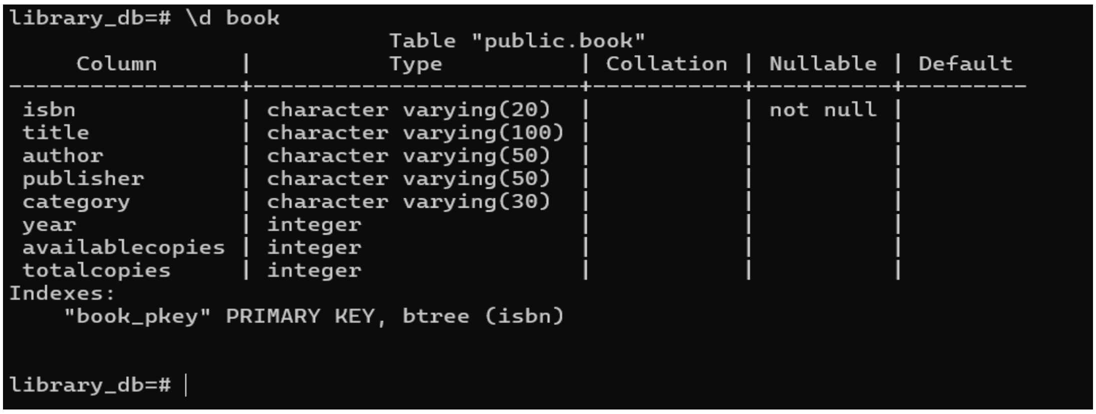
```
library_db=# \d Member
                        Table "public.member"
     Column     |          Type          | Collation | Nullable | Default
----------------+------------------------+-----------+----------+---------
 memberid       | integer                |           | not null |
 fname          | character varying(50)  |           |          |
 lname          | character varying(50)  |           |          |
 address        | character varying(100) |           |          |
 membershipdate | date                   |           |          |
Indexes:
    "member_pkey" PRIMARY KEY, btree (memberid)


library_db=#
```

3. Book Table



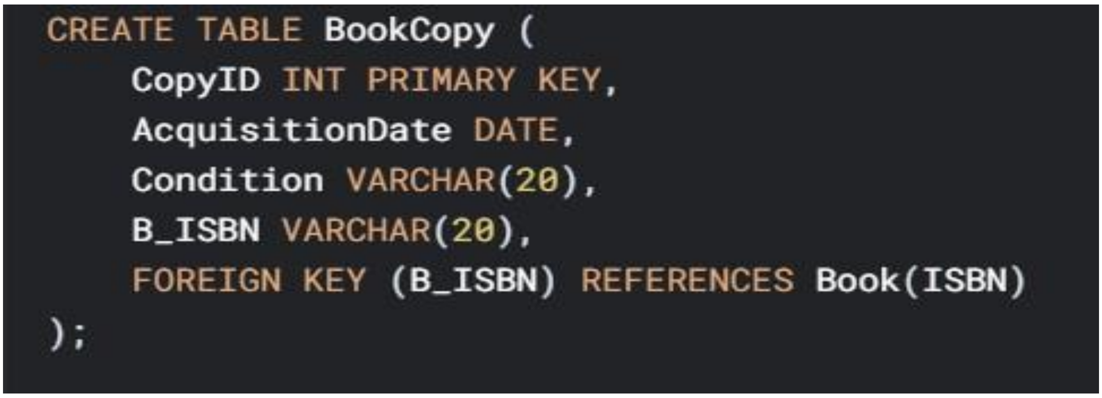
```
CREATE TABLE Book (
      ISBN VARCHAR(20) PRIMARY KEY,
      Title VARCHAR(100),
      Author VARCHAR(50),
      Publisher VARCHAR(50),
      Category VARCHAR(30),
      Year INT,
      TotalCopies INT,
      AvailableCopies INT
);
```



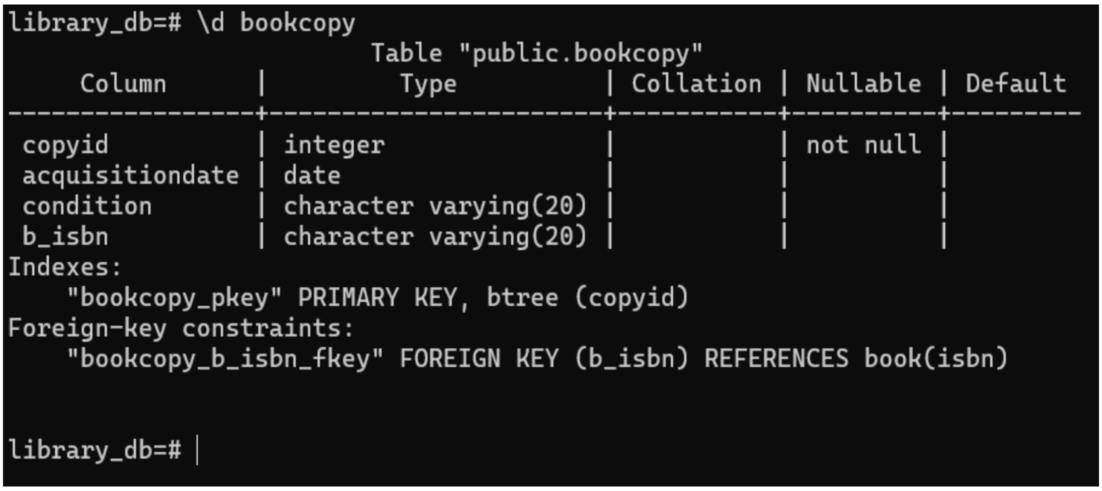
```
library_db=# \d book
                            Table "public.book"
     Column      |          Type          | Collation | Nullable | Default
-----------------+------------------------+-----------+----------+---------
 isbn            | character varying(20)  |           | not null |
 title           | character varying(100) |           |          |
 author          | character varying(50)  |           |          |
 publisher       | character varying(50)  |           |          |
 category        | character varying(30)  |           |          |
 year            | integer                |           |          |
 availablecopies | integer                |           |          |
 totalcopies     | integer                |           |          |
Indexes:
    "book_pkey" PRIMARY KEY, btree (isbn)


library_db=# 
```

4. BookCopy Table



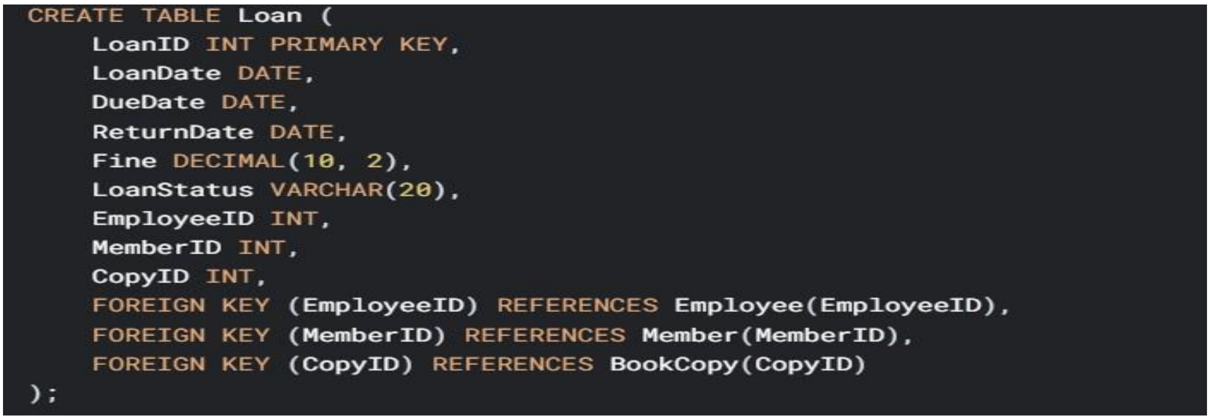
```
CREATE TABLE BookCopy (
    CopyID INT PRIMARY KEY,
    AcquisitionDate DATE,
    Condition VARCHAR(20),
    B_ISBN VARCHAR(20),
    FOREIGN KEY (B_ISBN) REFERENCES Book(ISBN)
);
```



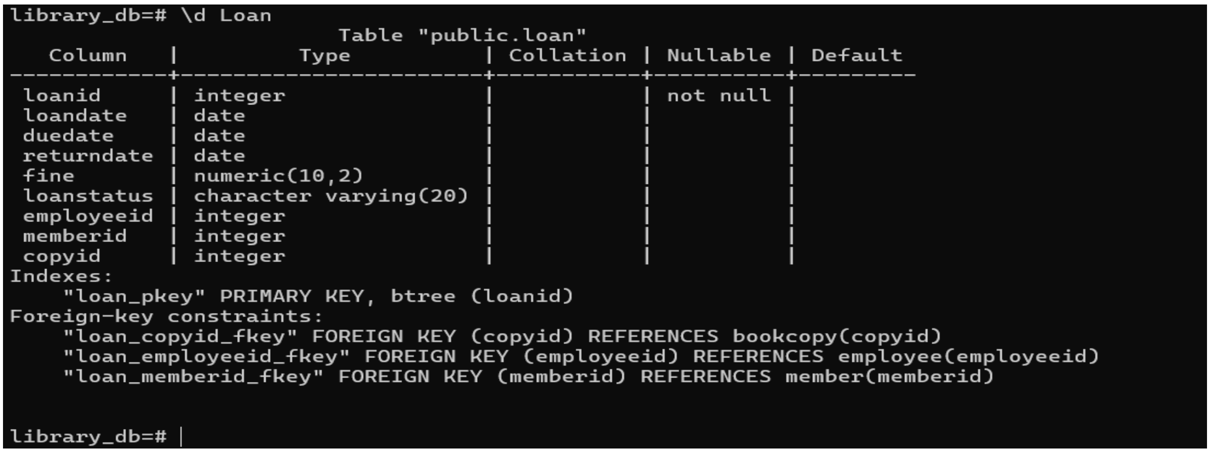
```
library_db=# \d bookcopy
                          Table "public.bookcopy"
     Column      |         Type          | Collation | Nullable | Default
-----------------+-----------------------+-----------+----------+---------
 copyid          | integer               |           | not null |
 acquisitiondate | date                  |           |          |
 condition       | character varying(20) |           |          |
 b_isbn          | character varying(20) |           |          |
Indexes:
    "bookcopy_pkey" PRIMARY KEY, btree (copyid)
Foreign-key constraints:
    "bookcopy_b_isbn_fkey" FOREIGN KEY (b_isbn) REFERENCES book(isbn)


library_db=# 
```

5. Loan Table



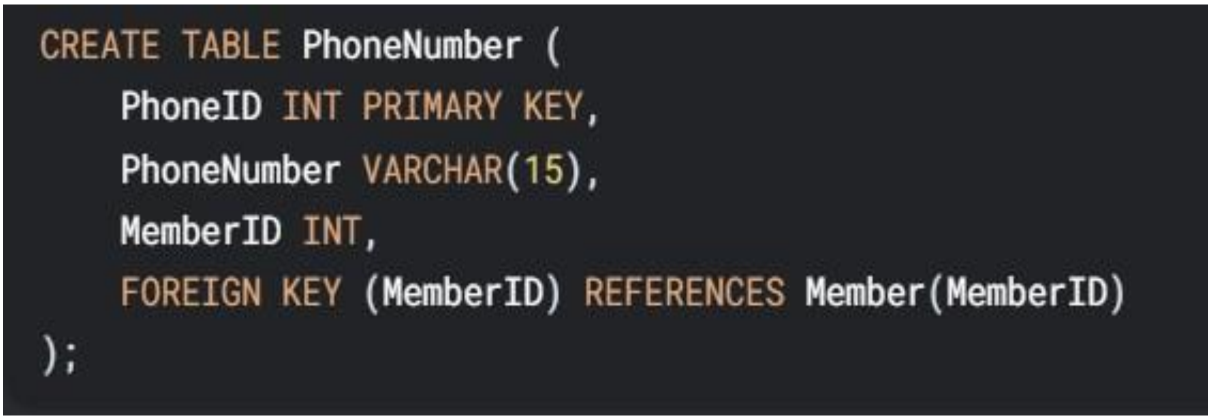
```
CREATE TABLE Loan (
    LoanID INT PRIMARY KEY,
    LoanDate DATE,
    DueDate DATE,
    ReturnDate DATE,
    Fine DECIMAL(10, 2),
    LoanStatus VARCHAR(20),
    EmployeeID INT,
    MemberID INT,
    CopyID INT,
    FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),
    FOREIGN KEY (MemberID) REFERENCES Member(MemberID),
    FOREIGN KEY (CopyID) REFERENCES BookCopy(CopyID)
);
```



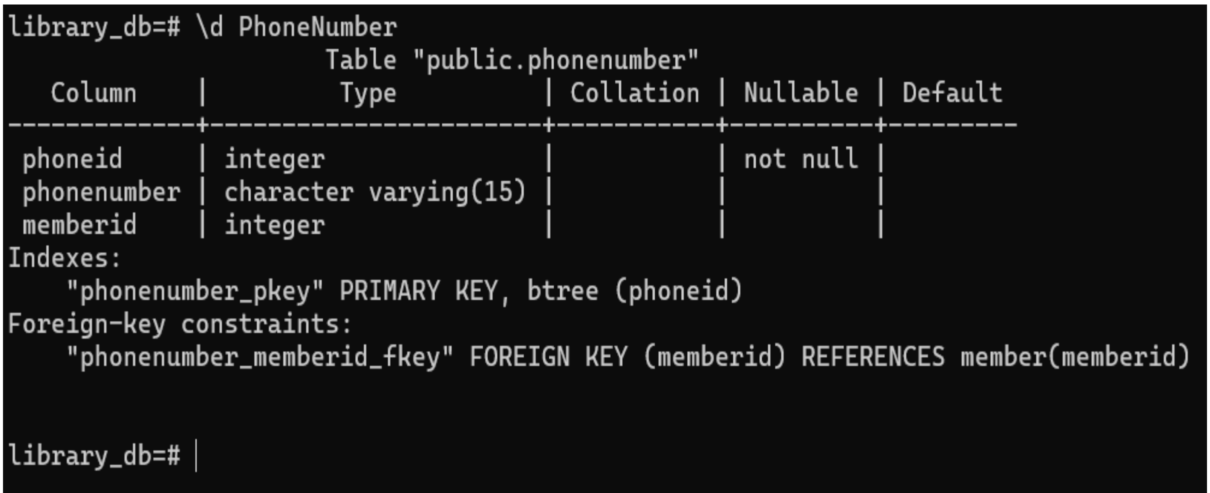
```
library_db=# \d Loan
                        Table "public.loan"
   Column    |         Type          | Collation | Nullable | Default
-------------+-----------------------+-----------+----------+---------
 loanid      | integer               |           | not null |
 loandate    | date                  |           |          |
 duedate     | date                  |           |          |
 returndate  | date                  |           |          |
 fine        | numeric(10,2)         |           |          |
 loanstatus  | character varying(20) |           |          |
 employeeid  | integer               |           |          |
 memberid    | integer               |           |          |
 copyid      | integer               |           |          |
Indexes:
    "loan_pkey" PRIMARY KEY, btree (loanid)
Foreign-key constraints:
    "loan_copyid_fkey" FOREIGN KEY (copyid) REFERENCES bookcopy(copyid)
    "loan_employeeid_fkey" FOREIGN KEY (employeeid) REFERENCES employee(employeeid)
    "loan_memberid_fkey" FOREIGN KEY (memberid) REFERENCES member(memberid)


library_db=# |
```

6.PhoneNumber Table



```
CREATE TABLE PhoneNumber (
    PhoneID INT PRIMARY KEY,
    PhoneNumber VARCHAR(15),
    MemberID INT,
    FOREIGN KEY (MemberID) REFERENCES Member(MemberID)
);
```

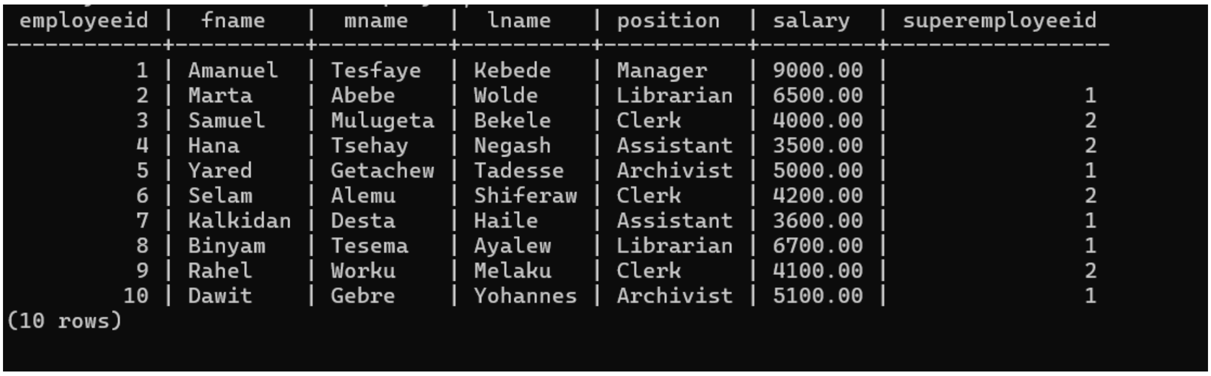


```
library_db=# \d PhoneNumber
                     Table "public.phonenumber"
   Column    |         Type          | Collation | Nullable | Default
-------------+-----------------------+-----------+----------+---------
 phoneid     | integer               |           | not null |
 phonenumber | character varying(15) |           |          |
 memberid    | integer               |           |          |
Indexes:
    "phonenumber_pkey" PRIMARY KEY, btree (phoneid)
Foreign-key constraints:
    "phonenumber_memberid_fkey" FOREIGN KEY (memberid) REFERENCES member(memberid)


library_db=# |
```

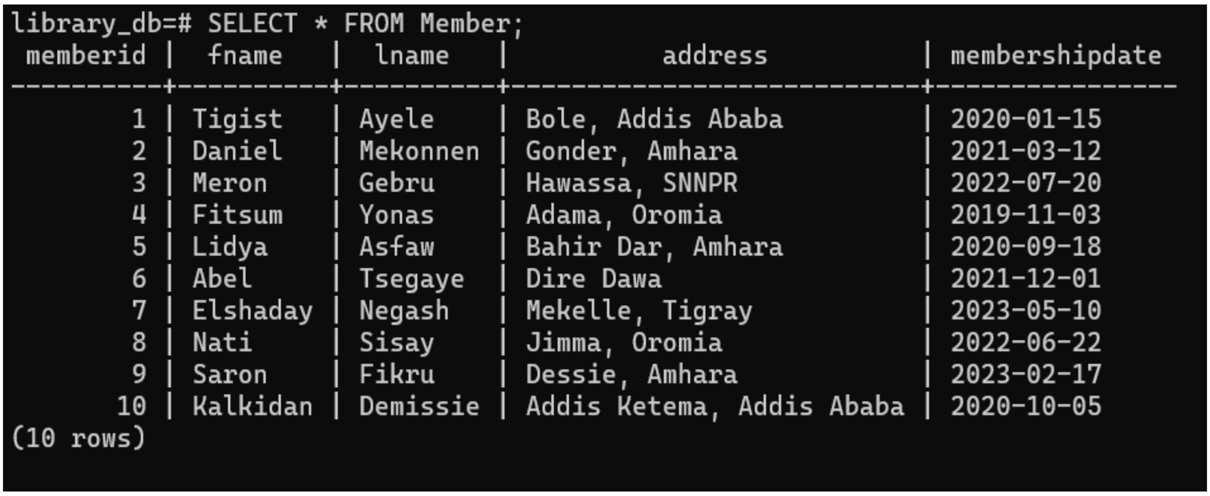
9. Now we can populate the set tables with the appropriate data to see how actual customer data is handled by the database.

1. Employee Table



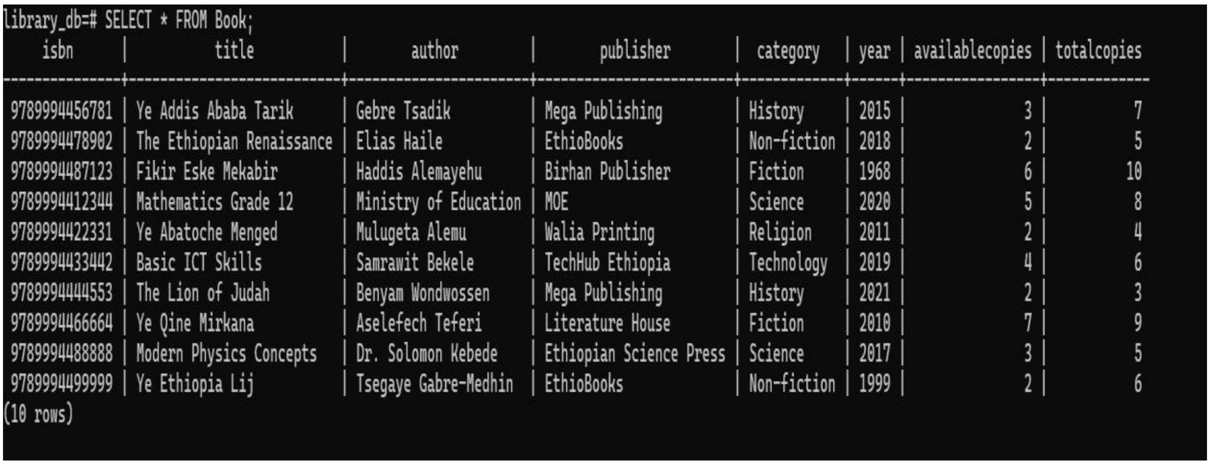
```
 employeeid |  fname   |  mname   |  lname   | position  | salary  | superemployeeid
------------+----------+----------+----------+-----------+---------+-----------------
          1 | Amanuel  | Tesfaye  | Kebede   | Manager   | 9000.00 |
          2 | Marta    | Abebe    | Wolde    | Librarian | 6500.00 |               1
          3 | Samuel   | Mulugeta | Bekele   | Clerk     | 4000.00 |               2
          4 | Hana     | Tsehay   | Negash   | Assistant | 3500.00 |               2
          5 | Yared    | Getachew | Tadesse  | Archivist | 5000.00 |               1
          6 | Selam    | Alemu    | Shiferaw | Clerk     | 4200.00 |               2
          7 | Kalkidan | Desta    | Haile    | Assistant | 3600.00 |               1
          8 | Binyam   | Tesema   | Ayalew   | Librarian | 6700.00 |               1
          9 | Rahel    | Worku    | Melaku   | Clerk     | 4100.00 |               2
         10 | Dawit    | Gebre    | Yohannes | Archivist | 5100.00 |               1
(10 rows)
```

2. Member Table



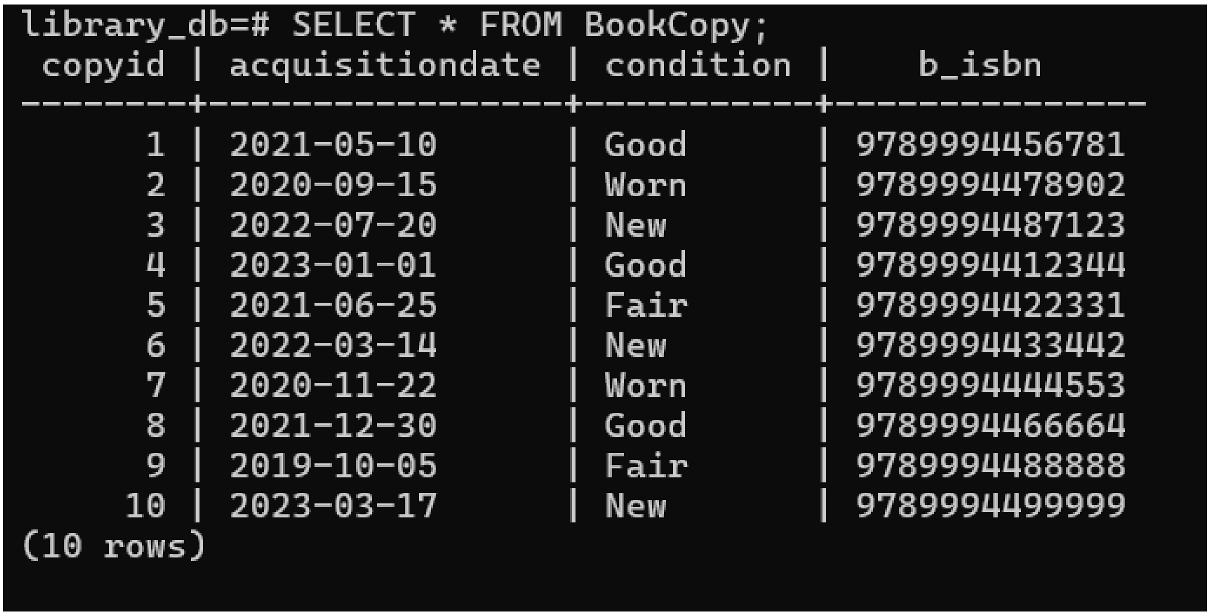
```
library_db=# SELECT * FROM Member;
 memberid |  fname   |  lname   |          address          | membershipdate
----------+----------+----------+---------------------------+----------------
        1 | Tigist   | Ayele    | Bole, Addis Ababa         | 2020-01-15
        2 | Daniel   | Mekonnen | Gonder, Amhara            | 2021-03-12
        3 | Meron    | Gebru    | Hawassa, SNNPR            | 2022-07-20
        4 | Fitsum   | Yonas    | Adama, Oromia             | 2019-11-03
        5 | Lidya    | Asfaw    | Bahir Dar, Amhara         | 2020-09-18
        6 | Abel     | Tsegaye  | Dire Dawa                 | 2021-12-01
        7 | Elshaday | Negash   | Mekelle, Tigray           | 2023-05-10
        8 | Nati     | Sisay    | Jimma, Oromia             | 2022-06-22
        9 | Saron    | Fikru    | Dessie, Amhara            | 2023-02-17
       10 | Kalkidan | Demissie | Addis Ketema, Addis Ababa | 2020-10-05
(10 rows)
```

3. Book Table



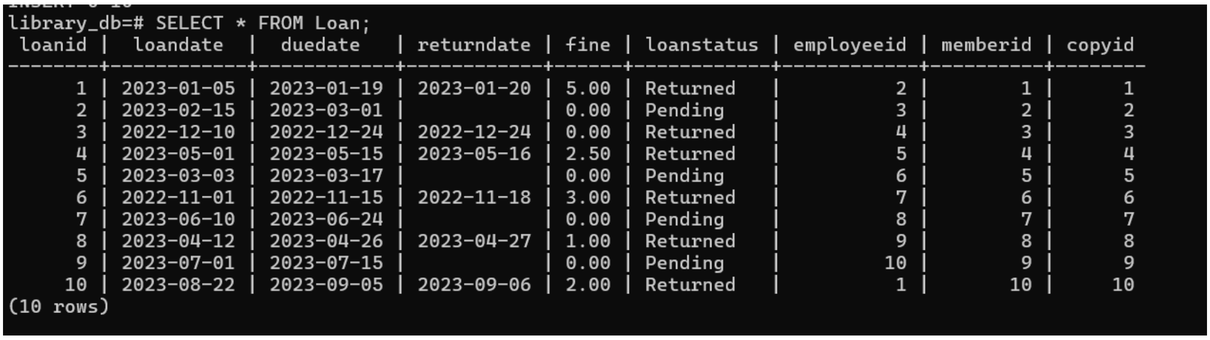
```
library_db=# SELECT * FROM Book;
     isbn      |           title           |        author         |        publisher        |  category   | year | availablecopies | totalcopies
---------------+---------------------------+-----------------------+-------------------------+-------------+------+-----------------+-------------
 9789994456781 | Ye Addis Ababa Tarik      | Gebre Tsadik          | Mega Publishing         | History     | 2015 |               3 |           7
 9789994478902 | The Ethiopian Renaissance | Elias Haile           | EthioBooks              | Non-fiction | 2018 |               2 |           5
 9789994487123 | Fikir Eske Mekabir        | Haddis Alemayehu      | Birhan Publisher        | Fiction     | 1968 |               6 |          10
 9789994412344 | Mathematics Grade 12      | Ministry of Education | MOE                     | Science     | 2020 |               5 |           8
 9789994422331 | Ye Abatoche Menged        | Mulugeta Alemu        | Walia Printing          | Religion    | 2011 |               2 |           4
 9789994433442 | Basic ICT Skills          | Samrawit Bekele       | TechHub Ethiopia        | Technology  | 2019 |               4 |           6
 9789994444553 | The Lion of Judah         | Benyam Wondwossen     | Mega Publishing         | History     | 2021 |               2 |           3
 9789994466664 | Ye Qine Mirkana           | Aselefech Teferi      | Literature House        | Fiction     | 2010 |               7 |           9
 9789994488888 | Modern Physics Concepts   | Dr. Solomon Kebede    | Ethiopian Science Press | Science     | 2017 |               3 |           5
 9789994499999 | Ye Ethiopia Lij           | Tsegaye Gabre-Medhin  | EthioBooks              | Non-fiction | 1999 |               2 |           6
(10 rows)
```

4. BooKCopy



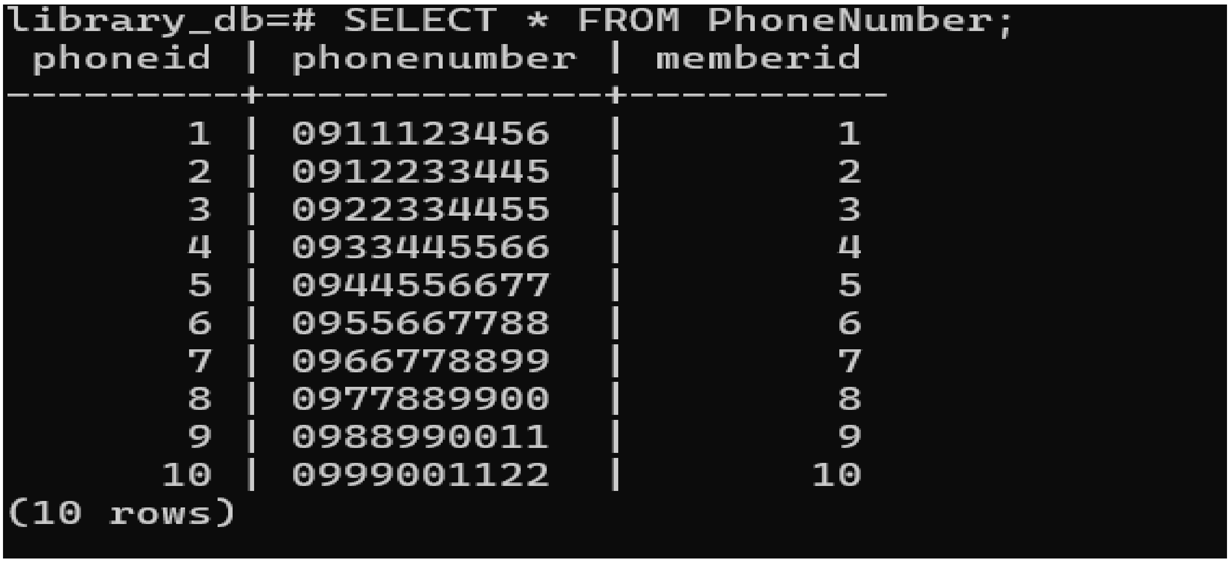
```
library_db=# SELECT * FROM BookCopy;
 copyid | acquisitiondate | condition |     b_isbn
--------+-----------------+-----------+---------------
      1 | 2021-05-10      | Good      | 9789994456781
      2 | 2020-09-15      | Worn      | 9789994478902
      3 | 2022-07-20      | New       | 9789994487123
      4 | 2023-01-01      | Good      | 9789994412344
      5 | 2021-06-25      | Fair      | 9789994422331
      6 | 2022-03-14      | New       | 9789994433442
      7 | 2020-11-22      | Worn      | 9789994444553
      8 | 2021-12-30      | Good      | 9789994466664
      9 | 2019-10-05      | Fair      | 9789994488888
     10 | 2023-03-17      | New       | 9789994499999
(10 rows)
```

5. Loan Table



```
INSERT 0 10
library_db=# SELECT * FROM Loan;
 loanid |  loandate  |  duedate   | returndate | fine | loanstatus | employeeid | memberid | copyid
--------+------------+------------+------------+------+------------+------------+----------+--------
      1 | 2023-01-05 | 2023-01-19 | 2023-01-20 | 5.00 | Returned   |          2 |        1 |      1
      2 | 2023-02-15 | 2023-03-01 |            | 0.00 | Pending    |          3 |        2 |      2
      3 | 2022-12-10 | 2022-12-24 | 2022-12-24 | 0.00 | Returned   |          4 |        3 |      3
      4 | 2023-05-01 | 2023-05-15 | 2023-05-16 | 2.50 | Returned   |          5 |        4 |      4
      5 | 2023-03-03 | 2023-03-17 |            | 0.00 | Pending    |          6 |        5 |      5
      6 | 2022-11-01 | 2022-11-15 | 2022-11-18 | 3.00 | Returned   |          7 |        6 |      6
      7 | 2023-06-10 | 2023-06-24 |            | 0.00 | Pending    |          8 |        7 |      7
      8 | 2023-04-12 | 2023-04-26 | 2023-04-27 | 1.00 | Returned   |          9 |        8 |      8
      9 | 2023-07-01 | 2023-07-15 |            | 0.00 | Pending    |         10 |        9 |      9
     10 | 2023-08-22 | 2023-09-05 | 2023-09-06 | 2.00 | Returned   |          1 |       10 |     10
(10 rows)
```

6. PhoneNumber Tabl

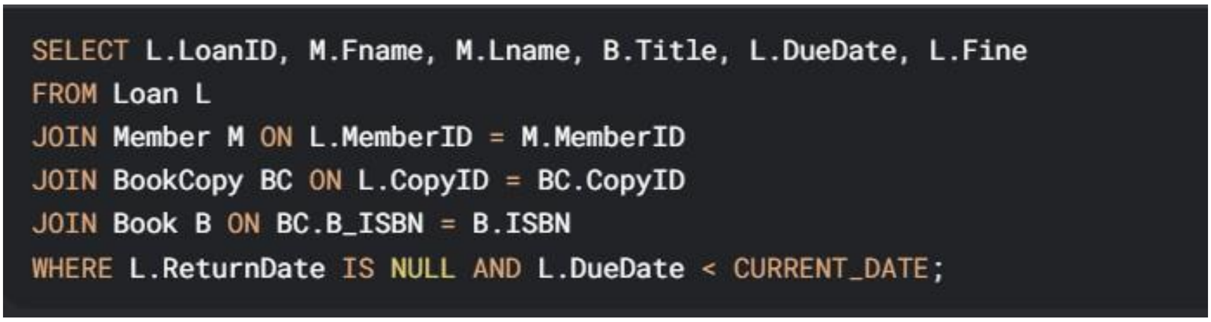


```
library_db=# SELECT * FROM PhoneNumber;
 phoneid | phonenumber | memberid
---------+-------------+----------
       1 | 0911123456  |        1
       2 | 0912233445  |        2
       3 | 0922334455  |        3
       4 | 0933445566  |        4
       5 | 0944556677  |        5
       6 | 0955667788  |        6
       7 | 0966778899  |        7
       8 | 0977889900  |        8
       9 | 0988990011  |        9
      10 | 0999001122  |       10
(10 rows)
```

10. Queries

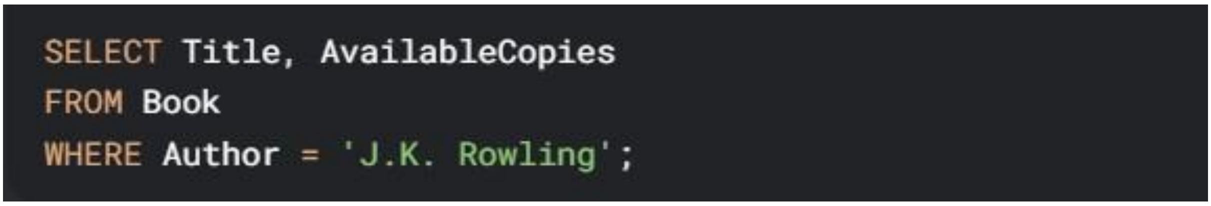
This subtopic details the relational algebra expressions for various SQL queries related to a library database. The queries include operations like retrieval, updates, deletions, and insertions across different tables.

1. **List all overdue books with member details:**



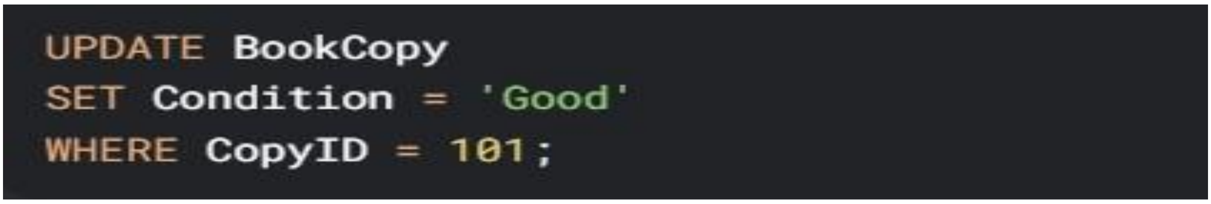
```
SELECT L.LoanID, M.Fname, M.Lname, B.Title, L.DueDate, L.Fine
FROM Loan L
JOIN Member M ON L.MemberID = M.MemberID
JOIN BookCopy BC ON L.CopyID = BC.CopyID
JOIN Book B ON BC.B_ISBN = B.ISBN
WHERE L.ReturnDate IS NULL AND L.DueDate < CURRENT_DATE;
```

2. Find books by a specific author:



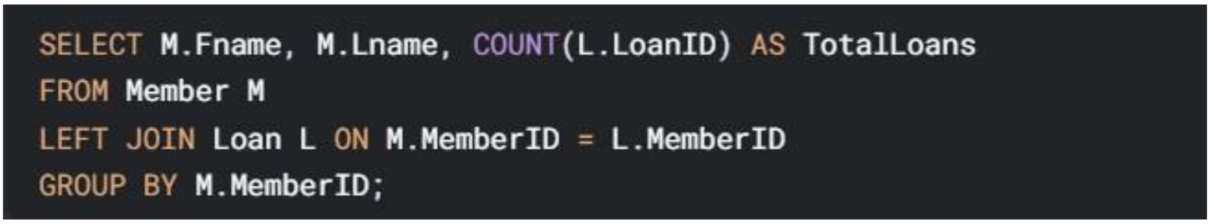
```
SELECT Title, AvailableCopies
FROM Book
WHERE Author = 'J.K. Rowling';
```

3. Update a book’s condition:



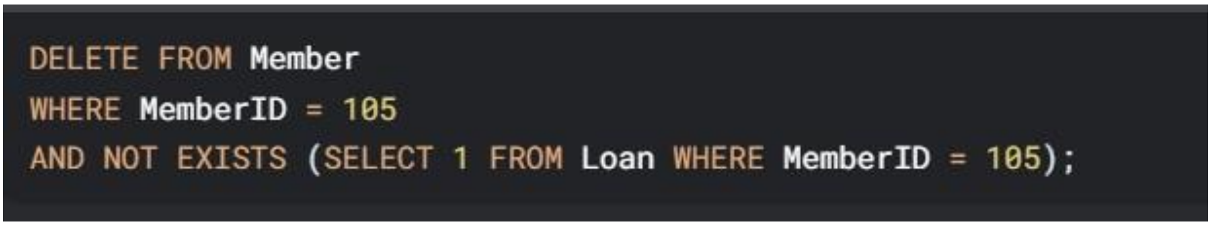
```
UPDATE BookCopy
SET Condition = 'Good'
WHERE CopyID = 101;
```

4. Count loans per member:



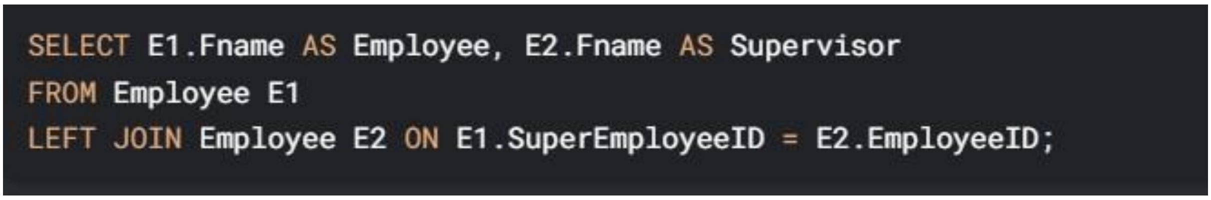
```
SELECT M.Fname, M.Lname, COUNT(L.LoanID) AS TotalLoans
FROM Member M
LEFT JOIN Loan L ON M.MemberID = L.MemberID
GROUP BY M.MemberID;
```

5. Delete an inactive member:



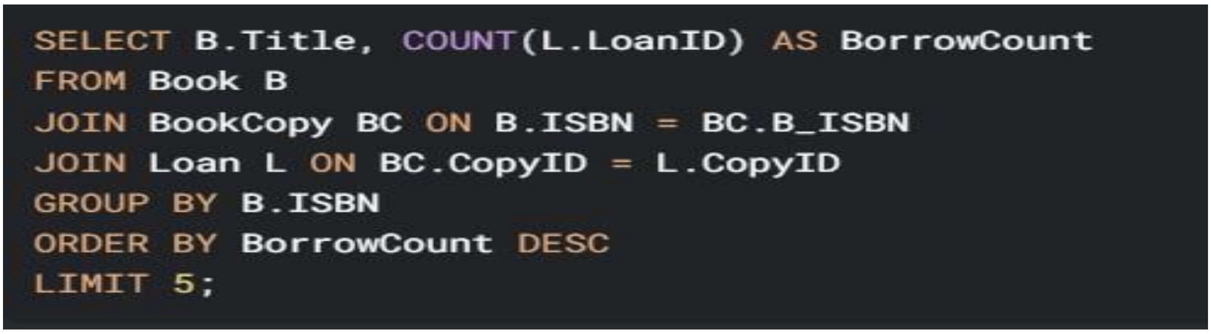
```
DELETE FROM Member
WHERE MemberID = 105
AND NOT EXISTS (SELECT 1 FROM Loan WHERE MemberID = 105);
```

6. List employees and their supervisors:



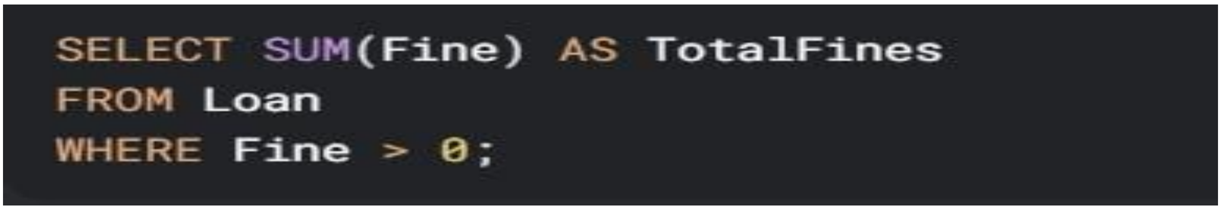
```
SELECT E1.Fname AS Employee, E2.Fname AS Supervisor
FROM Employee E1
LEFT JOIN Employee E2 ON E1.SuperEmployeeID = E2.EmployeeID;
```

7. Find most borrowed books:



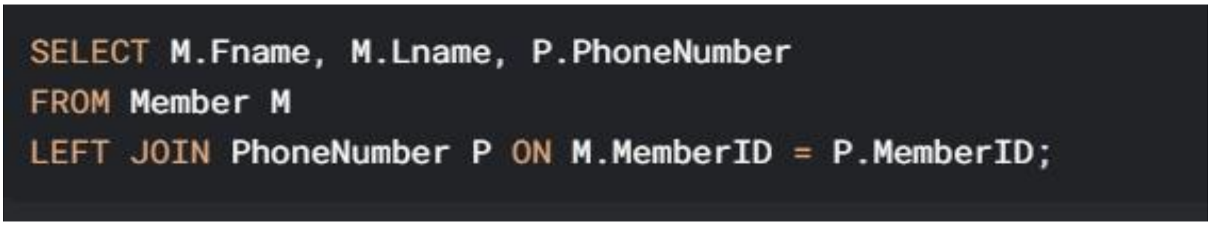
```
SELECT B.Title, COUNT(L.LoanID) AS BorrowCount
FROM Book B
JOIN BookCopy BC ON B.ISBN = BC.B_ISBN
JOIN Loan L ON BC.CopyID = L.CopyID
GROUP BY B.ISBN
ORDER BY BorrowCount DESC
LIMIT 5;
```

8. Calculate total fines collected:



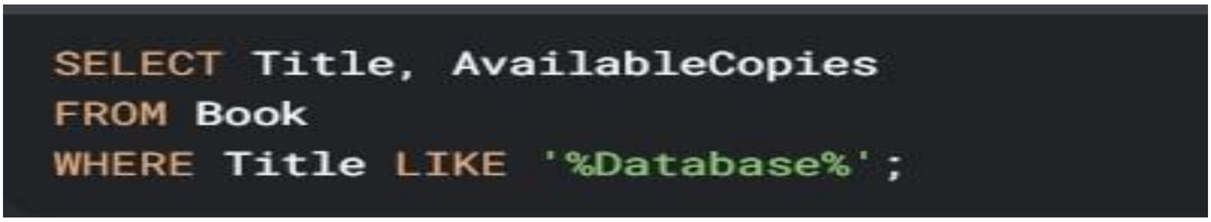
```
SELECT SUM(Fine) AS TotalFines
FROM Loan
WHERE Fine > 0;
```

9. List all members with phone numbers:



```
SELECT M.Fname, M.Lname, P.PhoneNumber
FROM Member M
LEFT JOIN PhoneNumber P ON M.MemberID = P.MemberID;
```

10. Check available copies of a book:



```
SELECT Title, AvailableCopies
FROM Book
WHERE Title LIKE '%Database%';
```