**SQL commands for table creation:**

-- Table for Locations

CREATE TABLE Location (LocationID NUMBER PRIMARY KEY,FloorNumber NUMBER CHECK (FloorNumber >= 0), ShelfNumber NUMBER CHECK (ShelfNumber >= 0));

-- Table for Library Resources

CREATE TABLE LibraryResource (ResourceID NUMBER PRIMARY KEY,Title VARCHAR2(200),ResourceType VARCHAR2(50),ClassNumber NUMBER,DigitalCopiesAvailable NUMBER CHECK (DigitalCopiesAvailable >= 0),LocationID NUMBER,LoanPeriod NUMBER CHECK (LoanPeriod >= 0),CONSTRAINT fk\_resource\_location FOREIGN KEY (LocationID) REFERENCES Location(LocationID));

-- Table for Library Members

CREATE TABLE LibraryMember (MemberID NUMBER PRIMARY KEY,Name VARCHAR2(100),MemberType VARCHAR2(50),MaxLoansAllowed NUMBER CHECK (MaxLoansAllowed >= 0), CardIssueDate DATE,CardExpirationDate DATE);

-- Table for Reservations

CREATE TABLE Reservation (ReservationID NUMBER PRIMARY KEY,ReservationDate DATE,ReservationTime TIMESTAMP,FailureCount NUMBER CHECK (FailureCount >= 0), NotificationDate DATE,ResourceID NUMBER,MemberID NUMBER,CONSTRAINT fk\_reservation\_resource FOREIGN KEY (ResourceID) REFERENCES LibraryResource(ResourceID),CONSTRAINT fk\_reservation\_member FOREIGN KEY (MemberID) REFERENCES LibraryMember(MemberID));

-- Table for Loans

CREATE TABLE Loan (LoanID NUMBER PRIMARY KEY,ResourceID NUMBER,MemberID NUMBER,LoanDate DATE,ReturnDate DATE,CONSTRAINT fk\_loan\_resource FOREIGN KEY (ResourceID) REFERENCES LibraryResource(ResourceID),CONSTRAINT fk\_loan\_member FOREIGN KEY (MemberID) REFERENCES LibraryMember(MemberID));

-- Table for Fines

CREATE TABLE Fine (FineID NUMBER PRIMARY KEY,LoanID NUMBER,MemberID NUMBER,FineAmount NUMBER CHECK (FineAmount >= 0), IsPaid CHAR(1) CHECK (IsPaid IN ('Y', 'N')), CONSTRAINT fk\_fine\_loan FOREIGN KEY (LoanID) REFERENCES Loan(LoanID),CONSTRAINT fk\_fine\_member FOREIGN KEY (MemberID) REFERENCES LibraryMember(MemberID) );

**Data population in tables:**

--LibraryMember

INSERT INTO LibraryMember (MemberID, Name, MemberType, MaxLoansAllowed, CardIssueDate, CardExpirationDate)

VALUES (1, 'Alice Smith', 'Student', 5, TO\_DATE('2021-09-01', 'YYYY-MM-DD'), TO\_DATE('2024-09-01', 'YYYY-MM-DD'));

INSERT INTO LibraryMember (MemberID, Name, MemberType, MaxLoansAllowed, CardIssueDate, CardExpirationDate)

VALUES (2, 'John Doe', 'Staff', 10, TO\_DATE('2020-05-01', 'YYYY-MM-DD'), TO\_DATE('2025-05-01', 'YYYY-MM-DD'));

INSERT INTO LibraryMember (MemberID, Name, MemberType, MaxLoansAllowed, CardIssueDate, CardExpirationDate)

VALUES (3, 'Mary Johnson', 'Student', 5, TO\_DATE('2022-01-01', 'YYYY-MM-DD'), TO\_DATE('2025-01-01', 'YYYY-MM-DD'));

INSERT INTO LibraryMember (MemberID, Name, MemberType, MaxLoansAllowed, CardIssueDate, CardExpirationDate)

VALUES (4, 'Grant Permission', 'Staff', 10, TO\_DATE('2021-07-01', 'YYYY-MM-DD'), TO\_DATE('2026-07-01', 'YYYY-MM-DD'));

INSERT INTO LibraryMember (MemberID, Name, MemberType, MaxLoansAllowed, CardIssueDate, CardExpirationDate)

VALUES (5, 'Emma Davis', 'Student', 5, TO\_DATE('2022-08-01', 'YYYY-MM-DD'), TO\_DATE('2025-08-01', 'YYYY-MM-DD'));

INSERT INTO LibraryMember (MemberID, Name, MemberType, MaxLoansAllowed, CardIssueDate, CardExpirationDate)

VALUES (6, 'Robert Lee', 'Staff', 10, TO\_DATE('2020-04-01', 'YYYY-MM-DD'), TO\_DATE('2023-04-01', 'YYYY-MM-DD'));

INSERT INTO LibraryMember (MemberID, Name, MemberType, MaxLoansAllowed, CardIssueDate, CardExpirationDate)

VALUES (7, 'Justin Time', 'Student', 5, TO\_DATE('2023-01-01', 'YYYY-MM-DD'), TO\_DATE('2026-01-01', 'YYYY-MM-DD'));

INSERT INTO LibraryMember (MemberID, Name, MemberType, MaxLoansAllowed, CardIssueDate, CardExpirationDate)

VALUES (8, 'Phil Ter Data', 'Staff', 10, TO\_DATE('2022-02-01', 'YYYY-MM-DD'), TO\_DATE('2025-02-01', 'YYYY-MM-DD'));

INSERT INTO LibraryMember (MemberID, Name, MemberType, MaxLoansAllowed, CardIssueDate, CardExpirationDate)

VALUES (9, 'Norm Alized', 'Student', 5, TO\_DATE('2021-03-01', 'YYYY-MM-DD'), TO\_DATE('2024-03-01', 'YYYY-MM-DD'));

INSERT INTO LibraryMember (MemberID, Name, MemberType, MaxLoansAllowed, CardIssueDate, CardExpirationDate)

VALUES (10, 'William Young', 'Staff', 10, TO\_DATE('2020-12-01', 'YYYY-MM-DD'), TO\_DATE('2023-12-01', 'YYYY-MM-DD'));

--Location

INSERT INTO Location (LocationID,FloorNumber,ShelfNumber) VALUES (101,1,1);

INSERT INTO Location (LocationID,FloorNumber,ShelfNumber) VALUES (201,2,1);

INSERT INTO Location (LocationID,FloorNumber,ShelfNumber) VALUES (103,1,3);

INSERT INTO Location (LocationID,FloorNumber,ShelfNumber) VALUES (104,1,4);

INSERT INTO Location (LocationID,FloorNumber,ShelfNumber) VALUES (301,3,1);

INSERT INTO Location (LocationID,FloorNumber,ShelfNumber) VALUES (0,0,0);

--Resources

INSERT INTO LibraryResource (ResourceID, Title, ResourceType, ClassNumber, DigitalCopiesAvailable, LocationID, LoanPeriod)

VALUES (1001, 'GROUP by Regret, ORDER BY Poor Decisions', 'Book',10,0,101,14);

INSERT INTO LibraryResource (ResourceID, Title, ResourceType, ClassNumber, DigitalCopiesAvailable, LocationID, LoanPeriod)

VALUES (1002, 'The Very Hungry Caterpillar ', 'Book',11,0,201,14);

INSERT INTO LibraryResource (ResourceID, Title, ResourceType, ClassNumber, DigitalCopiesAvailable, LocationID, LoanPeriod)

VALUES (1003, 'Principia Mathematica', 'Book',12,0, 103, 0);

INSERT INTO LibraryResource (ResourceID, Title, ResourceType, ClassNumber, DigitalCopiesAvailable, LocationID, LoanPeriod)

VALUES (1004, 'The Concept of Law', 'eBook',13, 7, 0, 14);

INSERT INTO LibraryResource (ResourceID, Title, ResourceType, ClassNumber, DigitalCopiesAvailable, LocationID, LoanPeriod)

VALUES (1005, 'The Feynman Lectures on Physics', 'Book',14,0, 104, 3);

INSERT INTO LibraryResource (ResourceID, Title, ResourceType, ClassNumber, DigitalCopiesAvailable, LocationID, LoanPeriod)

VALUES (1006, 'The Selfish Gene', 'eBook', 15,10,0, 3);

INSERT INTO LibraryResource (ResourceID, Title, ResourceType, ClassNumber, DigitalCopiesAvailable, LocationID, LoanPeriod)

VALUES (1007, 'Laptop1', 'Laptop', 0,0,301, 0);

INSERT INTO LibraryResource (ResourceID, Title, ResourceType, ClassNumber, DigitalCopiesAvailable, LocationID, LoanPeriod)

VALUES (1008, 'The Joy of SQL', 'Book',10,0,101, 0);

--Loan

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10001, 1001, 3,TO\_DATE('2024-08-01', 'YYYY-MM-DD'), TO\_DATE('2024-08-16', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10002, 1002,3, TO\_DATE('2024-07-15', 'YYYY-MM-DD'), TO\_DATE('2024-07-29', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10003, 1002, 5, TO\_DATE('2024-08-05', 'YYYY-MM-DD'), TO\_DATE('2024-08-19', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10004, 1002, 6, TO\_DATE('2024-09-01', 'YYYY-MM-DD'), TO\_DATE('2024-09-15', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10005, 1004,1, TO\_DATE('2024-06-10', 'YYYY-MM-DD'), TO\_DATE('2024-06-23', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10006, 1001, 3, TO\_DATE('2024-05-20', 'YYYY-MM-DD'), TO\_DATE('2024-06-02', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10007, 1006, 2, TO\_DATE('2024-07-01', 'YYYY-MM-DD'), TO\_DATE('2024-07-04', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10008, 1004, 2, TO\_DATE('2024-08-10', 'YYYY-MM-DD'), TO\_DATE('2024-08-24', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10009, 1005, 2, TO\_DATE('2024-09-05', 'YYYY-MM-DD'), TO\_DATE('2024-09-10', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10010, 1005, 10, TO\_DATE('2024-05-25', 'YYYY-MM-DD'), TO\_DATE('2024-05-28', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10011,1004,7, TO\_DATE('2024-06-15', 'YYYY-MM-DD'), TO\_DATE('2024-06-29', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10012, 1004, 7,TO\_DATE('2024-07-20', 'YYYY-MM-DD'), TO\_DATE('2024-08-03', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10013, 1004, 8, TO\_DATE('2024-08-25', 'YYYY-MM-DD'), TO\_DATE('2024-09-08', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10014,1004,3, TO\_DATE('2024-11-26', 'YYYY-MM-DD'), NULL);

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10015, 1002,1, TO\_DATE('2024-11-20', 'YYYY-MM-DD'),NULL);

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10016, 1006, 5, TO\_DATE('2024-07-02', 'YYYY-MM-DD'), TO\_DATE('2024-07-05', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10017, 1005, 9, TO\_DATE('2024-10-02', 'YYYY-MM-DD'), TO\_DATE('2024-10-30', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10018, 1001, 8, TO\_DATE('2024-11-08', 'YYYY-MM-DD'), TO\_DATE('2024-11-16', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10019, 1001, 4, TO\_DATE('2024-11-16', 'YYYY-MM-DD'), NULL);

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10020, 1005, 4, TO\_DATE('2024-10-30', 'YYYY-MM-DD'), TO\_DATE('2024-11-02', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10021, 1001, 4, TO\_DATE('2024-10-25', 'YYYY-MM-DD'), TO\_DATE('2024-11-08', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10022, 1005, 6, TO\_DATE('2024-11-02', 'YYYY-MM-DD'), TO\_DATE('2024-11-06', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10023, 1005, 2, TO\_DATE('2024-11-06', 'YYYY-MM-DD'), TO\_DATE('2024-11-10', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10024, 1005, 9, TO\_DATE('2024-11-10', 'YYYY-MM-DD'), TO\_DATE('2024-11-13', 'YYYY-MM-DD'));

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10025, 1004, 6, TO\_DATE('2024-11-27', 'YYYY-MM-DD'),NULL);

INSERT INTO Loan (LoanID, ResourceID, MemberID, LoanDate, ReturnDate)

VALUES (10026, 1005, 2, TO\_DATE('2024-11-22', 'YYYY-MM-DD'),NULL);

--Fine

INSERT INTO Fine (FineID,LoanID,MemberID,FineAmount,IsPaid) VALUES (1,10001,3,1.00,'Y');

INSERT INTO Fine (FineID,LoanID,MemberID,FineAmount,IsPaid) VALUES (2,10009,2,2.00,'N');

INSERT INTO Fine (FineID,LoanID,MemberID,FineAmount,IsPaid) VALUES (3,10017,3,25.00,'N');

INSERT INTO Fine (FineID,LoanID,MemberID,FineAmount,IsPaid) VALUES (4,10022,6,1.00,'Y');

INSERT INTO Fine (FineID,LoanID,MemberID,FineAmount,IsPaid) VALUES (5,10023,2,1.00,'N');

--Reservations

INSERT INTO Reservation (ReservationID, ReservationDate, ReservationTime, FailureCount, NotificationDate, ResourceID, MemberID)

VALUES (1, TO\_DATE('2024-11-01', 'YYYY-MM-DD'), TO\_TIMESTAMP('2024-11-01 09:00:00', 'YYYY-MM-DD HH24:MI:SS'), 3, TO\_DATE('2024-11-09', 'YYYY-MM-DD'), 1005, 7);

INSERT INTO Reservation (ReservationID, ReservationDate, ReservationTime, FailureCount, NotificationDate, ResourceID, MemberID)

VALUES (2, TO\_DATE('2024-11-01', 'YYYY-MM-DD'), TO\_TIMESTAMP('2024-11-01 11:00:00', 'YYYY-MM-DD HH24:MI:SS'), 0, TO\_DATE('2024-11-01', 'YYYY-MM-DD'), 1005, 6);

INSERT INTO Reservation (ReservationID, ReservationDate, ReservationTime, FailureCount, NotificationDate, ResourceID, MemberID)

VALUES (3, TO\_DATE('2024-11-03', 'YYYY-MM-DD'), TO\_TIMESTAMP('2024-11-03 14:00:00', 'YYYY-MM-DD HH24:MI:SS'), 1, TO\_DATE('2024-11-09', 'YYYY-MM-DD'), 1005, 9);

INSERT INTO Reservation (ReservationID, ReservationDate, ReservationTime, FailureCount, NotificationDate, ResourceID, MemberID)

VALUES (4, TO\_DATE('2024-11-04', 'YYYY-MM-DD'), TO\_TIMESTAMP('2024-11-04 18:27:01', 'YYYY-MM-DD HH24:MI:SS'), 0, TO\_DATE('2024-11-05', 'YYYY-MM-DD'), 1005, 2);

INSERT INTO Reservation (ReservationID, ReservationDate, ReservationTime, FailureCount, NotificationDate, ResourceID, MemberID)

VALUES (5, TO\_DATE('2024-11-27', 'YYYY-MM-DD'), TO\_TIMESTAMP('2024-11-27 19:00:00', 'YYYY-MM-DD HH24:MI:SS'), 0,NULL, 1002, 6);

INSERT INTO Reservation (ReservationID, ReservationDate, ReservationTime, FailureCount, NotificationDate, ResourceID, MemberID)

VALUES (6, TO\_DATE('2024-11-05', 'YYYY-MM-DD'), TO\_TIMESTAMP('2024-11-05 12:27:16', 'YYYY-MM-DD HH24:MI:SS'), 0, TO\_DATE('2024-11-07', 'YYYY-MM-DD'), 1001, 8);

INSERT INTO Reservation (ReservationID, ReservationDate, ReservationTime, FailureCount, NotificationDate, ResourceID, MemberID)

VALUES (7, TO\_DATE('2024-11-13', 'YYYY-MM-DD'), TO\_TIMESTAMP('2024-11-13 13:31:21', 'YYYY-MM-DD HH24:MI:SS'), 0, TO\_DATE('2024-11-16', 'YYYY-MM-DD'), 1001, 4);

INSERT INTO Reservation (ReservationID, ReservationDate, ReservationTime, FailureCount, NotificationDate, ResourceID, MemberID)

VALUES (8, TO\_DATE('2024-11-22', 'YYYY-MM-DD'), TO\_TIMESTAMP('2024-11-22 09:00:00', 'YYYY-MM-DD HH24:MI:SS'), 0,NULL, 1001, 3);

**SQL views:**

/\* This view provides a summary of all current loans, indicating whether they are overdue. \*/

CREATE OR REPLACE VIEW v\_CurrentLoans AS SELECT L.LoanID,L.ResourceID,M.Name AS MemberName,L.LoanDate,CASE WHEN L.ReturnDate IS NULL AND L.LoanDate+R.LoanPeriod < SYSDATE THEN 'Yes' ELSE 'No' END AS IsOverdue,L.LoanDate+R.LoanPeriod AS DateToReturn,R.title AS ResourceTitle FROM Loan L JOIN LibraryMember M ON L.MemberID=M.MemberID JOIN LibraryResource R ON L.ResourceID=R.ResourceID WHERE L.ReturnDate IS NULL;

/\* This view summarises the activity of each library member, detailing their current loans, reservations, any outstanding fines and suspension status. \*/

CREATE OR REPLACE VIEW v\_MemberActivityOverview AS SELECT M.MemberID,M.Name,M.MemberType,COUNT(DISTINCT vL.LoanID) AS TotalCurrentLoans,COUNT(DISTINCT R.ReservationID) AS TotalReservations,SUM(CASE WHEN F.IsPaid='N' THEN F.FineAmount ELSE 0 END) AS CurrentTotalFine,CASE WHEN SUM(CASE WHEN F.IsPaid='N' THEN F.FineAmount ELSE 0 END)>10 THEN 'Suspended' ELSE 'Not Suspended' END AS SuspensionStatus,MAX(L.LoanDate+Rs.LoanPeriod) AS LatestDueDate,MAX(R.ReservationDate) AS LatestReservationDate FROM LibraryMember M LEFT JOIN Loan L ON M.MemberID=L.MemberID LEFT JOIN Reservation R ON M.MemberID=R.MemberID LEFT JOIN LibraryResource Rs ON L.ResourceID=Rs.ResourceID LEFT JOIN Fine F ON L.LoanID=F.LoanID LEFT JOIN v\_CurrentLoans vL ON L.LoanID=vL.LoanID GROUP BY M.MemberID,M.Name,M.MemberType;

/\* This view provides information about all resources, including their availability status and locations. \*/

CREATE OR REPLACE VIEW v\_ResourceAvailability AS SELECT R.ResourceID,R.title,R.ResourceType,R.DigitalCopiesAvailable,R.LocationID,CASE WHEN R.ResourceType!='Book' THEN R.DigitalCopiesAvailable-(SELECT COUNT(\*) FROM v\_CurrentLoans vL WHERE vL.ResourceID=R.ResourceID) ELSE NULL END AS AvailableCopies,CASE WHEN R.ResourceType!='Book' AND EXISTS (SELECT 1 FROM v\_CurrentLoans vL WHERE vL.ResourceID=R.ResourceID GROUP BY vL.ResourceID HAVING COUNT(\*)>=R.DigitalCopiesAvailable) THEN 'Not Available' WHEN R.ResourceType='Book' AND EXISTS (SELECT 1 FROM v\_CurrentLoans vL WHERE vL.ResourceID=R.ResourceID) THEN 'Not Available' ELSE 'Available' END AS Availability,(SELECT vL.DateToReturn FROM v\_CurrentLoans vL WHERE vL.ResourceID=R.ResourceID AND vL.DateToReturn IS NOT NULL FETCH FIRST 1 ROWS ONLY) AS ReturnDate,L.FloorNumber,L.ShelfNumber FROM LibraryResource R JOIN Location L ON R.LocationID=L.LocationID GROUP BY R.ResourceID,R.Title,R.ResourceType,R.DigitalCopiesAvailable,R.LocationID,L.FloorNumber,L.ShelfNumber;

/\*This view shows all outstanding fines owed by members, providing details on what caused the fine.\*/

CREATE OR REPLACE VIEW v\_FinesDue AS SELECT F.FineID,M.Name AS MemberName,F.LoanID,F.FineAmount,F.IsPaid,L.LoanDate,L.LoanDate+R.LoanPeriod AS DueDate,L.ReturnDate FROM Fine F JOIN Loan L ON F.LoanID=L.LoanID JOIN LibraryResource R ON L.ResourceID=R.ResourceID JOIN LibraryMember M ON L.MemberID=M.MemberID WHERE F.IsPaid='N';

**SQL queries:**

Simple queries:

1) Retrieve all student member names

SELECT Name,MemberType FROM LibraryMember WHERE MemberType='Student';A screenshot of a cellphone

Description automatically generated

2) Find resources with 14-day loan periods:

SELECT Title,LoanPeriod FROM LibraryResource WHERE LoanPeriod=14;A screenshot of a computer

Description automatically generated

3) Get titles and locations for books on the first floor

SELECT R.Title,L.FloorNumber,L.ShelfNumber FROM LibraryResource R JOIN Location L ON R.LocationID=L.LocationID WHERE L.FloorNumber=1 AND R.ResourceType='Book';A screenshot of a white sheet

Description automatically generated

4) Find members with fines over 5 Pounds

SELECT M.Name,F.FineAmount FROM LibraryMember M JOIN Fine F ON M.MemberID=F.MemberID WHERE F.FineAmount>5;

A screenshot of a white grid

Description automatically generated

Intermediate queries:  
1) List unpaid fine details by member, including member name, fine amount, and loan details

SELECT M.Name,F.FineAmount,F.LoanID,F.DueDate,F.ReturnDate FROM v\_FinesDue F JOIN LibraryMember M ON F.MemberName=M.Name;

A white rectangular grid with numbers and letters

Description automatically generated with medium confidence

2) Retrieve current loan count and suspension status for all library members

SELECT M.Name,MA.TotalCurrentLoans,MA.SuspensionStatus FROM LibraryMember M JOIN v\_MemberActivityOverview MA ON M.MemberID=MA.MemberID;A screenshot of a white document

Description automatically generated

3) List overdue loans including member name, resource title, and return date.

SELECT M.Name,CL.ResourceTitle,CL.DateToReturn FROM v\_CurrentLoans CL JOIN LibraryMember M ON CL.MemberName=M.Name WHERE CL.IsOverdue='Yes';A screenshot of a computer

Description automatically generated

4) Check resource availability including title, resource type, and availability status.

SELECT R.Title,R.ResourceType,VA.Availability FROM LibraryResource R JOIN v\_ResourceAvailability VA ON R.ResourceID=VA.ResourceID;

A screenshot of a computer

Description automatically generated

Advanced queries:

1) Retrieve the top 3 most popular library resources based on loan count

SELECT R.Title,COUNT(L.LoanID) AS TotalLoans FROM LibraryResource R JOIN Loan L ON R.ResourceID=L.ResourceID GROUP BY R.Title ORDER BY TotalLoans DESC FETCH FIRST 3 ROWS ONLY;A screenshot of a computer

Description automatically generated

2) Calculate total cost of fines generated for each library resource

SELECT R.Title,SUM(F.FineAmount) AS TotalFines FROM LibraryResource R JOIN Loan L ON R.ResourceID=L.ResourceID JOIN Fine F ON L.LoanID=F.LoanID GROUP BY R.Title;A screenshot of a cell phone

Description automatically generated

3) Summarise member loans and reservations for active members (more than one loan and at least one reservation)

SELECT M.Name,COUNT(DISTINCT L.LoanID) AS TotalLoans,COUNT(DISTINCT R.ReservationID) AS TotalReservations FROM LibraryMember M LEFT JOIN Loan L ON M.MemberID=L.MemberID LEFT JOIN Reservation R ON M.MemberID=R.MemberID GROUP BY M.Name HAVING COUNT(DISTINCT L.LoanID)>1 AND COUNT(DISTINCT R.ReservationID)>0;

A table with text on it

Description automatically generated

4) Analyse detailed member activity: total loans, overdue loans, active reservations, and unpaid fines

SELECT M.MemberID,M.Name,COUNT(DISTINCT L.LoanID) AS TotalLoans,SUM(CASE WHEN L.ReturnDate IS NULL AND L.LoanDate+R.LoanPeriod<CURRENT\_DATE THEN 1 ELSE 0 END) AS OverdueLoans,COUNT(DISTINCT CASE WHEN L.ReturnDate IS NULL THEN Re.ReservationID END) AS TotalCurrentReservations,SUM(CASE WHEN F.IsPaid='N' THEN F.FineAmount ELSE 0 END) AS TotalUnpaidFines FROM LibraryMember M LEFT JOIN Loan L ON M.MemberID=L.MemberID LEFT JOIN LibraryResource R ON L.ResourceID=R.ResourceID LEFT JOIN Fine F ON L.LoanID=F.LoanID LEFT JOIN Reservation Re ON M.MemberID=Re.MemberID GROUP BY M.MemberID,M.Name HAVING COUNT(DISTINCT L.LoanID)>0 OR COUNT(DISTINCT Re.ReservationID)>0 ORDER BY TotalUnpaidFines DESC,OverdueLoans DESC;

A screenshot of a white sheet

Description automatically generated

Disclaimers:

* Someone who is currently suspended could take out a book without any problems.
* People could also loan more books than they are allowed.
* People could technically take out a loan for a resource that is already on loan.

**Bibliography**

Connolly, T., & Begg, C. (2019). *Database systems: A practical approach to design, implementation, and management* (6th ed.). Pearson Education.