

**Assignment 2 - 26/08/2025**  
**Faiz Khan**

```
-- MEMBER table
CREATE TABLE Member (
    Member_Id          NUMBER(5)          PRIMARY KEY,
    Member_Name        VARCHAR2(30),
    Member_Address     VARCHAR2(50),
    Acc_Open_Date      DATE,
    Membership_Type    VARCHAR2(20),
    Fees_Paid          NUMBER(4),
    Max_Books_Allowed  NUMBER(2),
    Penalty_Amount     NUMBER(7,2)
);

-- BOOKS table
CREATE TABLE Books (
    Book_No            NUMBER(6)          PRIMARY KEY,
    Book_Name          VARCHAR2(30),
    Author_Name        VARCHAR2(30),
    Cost               NUMBER(7,2),
    Category           CHAR(10)
);

-- ISSUE table
CREATE TABLE Issue (
    Lib_Issue_Id       NUMBER(10)         PRIMARY KEY,
    Book_No            NUMBER(6),
    Member_Id          NUMBER(5),
    Issue_Date         DATE,
    Return_Date        DATE,
    CONSTRAINT fk_issue_book FOREIGN KEY (Book_No)
        REFERENCES Books(Book_No),
    CONSTRAINT fk_issue_member FOREIGN KEY (Member_Id)
        REFERENCES Member(Member_Id)
);

--Data population
--Member
INSERT INTO Member (Member_Id, Member_Name, Member_Address, Acc_Open_Date,
Membership_Type, Fees_Paid, Max_Books_Allowed, Penalty_Amount)
VALUES (101, 'Rajiv Kumar', 'Delhi', DATE '2006-03-15', 'Annual', 500, 5, 50);

INSERT INTO Member VALUES (102, 'Geeta Iyer', 'Mumbai', DATE '2006-07-10',
'Lifetime', 2000, 10, 0);
INSERT INTO Member VALUES (103, 'Amit Sharma', 'Kolkata', DATE '2005-12-05', 'Half
Yearly', 300, 3, 25);
INSERT INTO Member VALUES (104, 'Rohit Singh', 'Chennai', DATE '2007-01-22',
'Quarterly', 150, 2, 75);
INSERT INTO Member VALUES (105, 'Girish Rao', 'Bangalore', DATE '2006-08-01',
'Annual', 500, 5, 10);
INSERT INTO Member VALUES (106, 'Sonia Patel', 'Hyderabad', DATE '2008-09-14',
'Annual', 600, 4, 0);
INSERT INTO Member VALUES (107, 'Mohit Jain', 'Pune', DATE '2006-11-20',
'Lifetime', 2500, 12, 100);
```

```

INSERT INTO Member VALUES (108, 'Ritika Ghosh', 'Delhi', DATE '2009-05-30', 'Half
Yearly', 350, 3, 0);
INSERT INTO Member VALUES (109, 'Sunil Das', 'Jaipur', DATE '2006-02-12', 'Annual',
400, 4, 20);
INSERT INTO Member VALUES (110, 'Anita Gill', 'Lucknow', DATE '2006-12-25',
'Annual', 550, 5, 0);

--Books
INSERT INTO Books (Book_No, Book_Name, Author_Name, Cost, Category)
VALUES (1001, 'Oracle SQL Basics', 'Loni', 550, 'Database');

INSERT INTO Books VALUES (1002, 'Advanced Database Systems', 'Loni', 750,
'Database');
INSERT INTO Books VALUES (1003, 'Science for All', 'Verma', 450, 'Science');
INSERT INTO Books VALUES (1004, 'Fictional Dreams', 'Roy', 600, 'Fiction');
INSERT INTO Books VALUES (1005, 'SQL for Beginners', 'Sharma', 500, 'Database');
INSERT INTO Books VALUES (1006, 'Management 101', 'Kapoor', 700, 'Management');
INSERT INTO Books VALUES (1007, 'The RDBMS Guide', 'Sen', 800, 'RDBMS');
INSERT INTO Books VALUES (1008, 'Fun with Physics', 'Das', 650, 'Science');
INSERT INTO Books VALUES (1009, 'Modern Fiction', 'Loni', 300, 'Fiction');
INSERT INTO Books VALUES (1010, 'SQL Performance Tuning', 'Mehta', 900,
'Database');

--Issue
-- Member 101 issues
INSERT INTO Issue VALUES (7001, 1001, 101, DATE '2006-07-10', NULL); -- not
returned
INSERT INTO Issue VALUES (7002, 1002, 101, DATE '2006-07-15', DATE '2006-08-20'); -
- returned
INSERT INTO Issue VALUES (7003, 1003, 101, DATE '2006-06-01', DATE '2006-08-10'); -
- more than 30 days

-- Member 102 issues
INSERT INTO Issue VALUES (7004, 1004, 102, DATE '2006-07-20', DATE '2006-07-30');
INSERT INTO Issue VALUES (7005, 1005, 102, DATE '2006-08-05', NULL); -- excluded
from update
INSERT INTO Issue VALUES (7006, 1006, 102, DATE '2006-08-10', NULL); -- excluded
from update
INSERT INTO Issue VALUES (7007, 1007, 103, DATE '2006-09-01', DATE '2006-09-20');
INSERT INTO Issue VALUES (7008, 1008, 104, DATE '2006-10-01', NULL);
INSERT INTO Issue VALUES (7009, 1009, 105, DATE '2006-11-15', DATE '2006-12-10');
INSERT INTO Issue VALUES (7010, 1010, 106, DATE '2007-01-01', DATE '2007-01-25');
INSERT INTO Issue VALUES (7011, 1001, 107, DATE '2006-05-01', DATE '2006-05-20');
INSERT INTO Issue VALUES (7012, 1005, 109, DATE '2006-02-15', NULL);

--checking data
select * from MEMBER;
select * from BOOKS;
select * from ISSUE;

-- 1) List all the books that are written by Author Loni and has price less than
600.
SELECT * FROM Books
WHERE Author_Name = 'Loni' AND Cost < 600;

-- 2) List the Issue details for the books that are not returned yet.

```

```

SELECT * FROM Issue
WHERE Return_Date IS NULL;

-- 3) Update all the blank return_date with 31-Dec-06 excluding 7005 and 7006.
UPDATE Issue SET Return_Date = DATE '2006-12-31'
WHERE Return_Date IS NULL
      AND Lib_Issue_Id NOT IN (7005,7006);

-- 4) List all the Issue details that have books issued for more than 30 days.
SELECT * FROM Issue
WHERE Return_Date IS NOT NULL
      AND (Return_Date - Issue_Date) > 30;

-- 5) List all the books that have price in range of 500 to 750 and has category as
Database.
SELECT * FROM Books
WHERE Cost BETWEEN 500 AND 750 AND Category = 'Database';

-- 6) List all the books that belong to any one of the following categories:
Science, Database, Fiction, Management.
SELECT * FROM Books
WHERE Category IN ('Science','Database','Fiction','Management');

-- 7) List all the members in the descending order of Penalty due on them.
SELECT * FROM Member ORDER BY Penalty_Amount DESC;

-- 8) List all the books in ascending order of category and descending order of
price.
SELECT * FROM Books ORDER BY Category ASC, Cost DESC;

-- 9) List all the books that contain word SQL in the name of the book.
SELECT * FROM Books
WHERE UPPER(Book_Name) LIKE '%SQL%';

-- 10) List all the members whose name starts with R or G and contains letter I in
it.
SELECT * FROM Member
WHERE (Member_Name LIKE 'R%' OR Member_Name LIKE 'G%')
      AND INSTR(UPPER(Member_Name), 'I') > 0;

-- 11) List the entire book name in Init cap and author in upper case in the
descending order of the book name.
SELECT INITCAP(Book_Name) AS Book_Name, UPPER(Author_Name) AS Author_Name
FROM Books
ORDER BY Book_Name DESC;

-- 12) List the Issue Details for all the books issued by member 101 with
Issue_date and Return Date in format: 'Monday, July, 10, 2006'.
SELECT Lib_Issue_Id, Book_No, Member_Id,
      TO_CHAR(Issue_Date, 'Day, Month, DD, YYYY') AS Issue_Date,
      TO_CHAR(Return_Date, 'Day, Month, DD, YYYY') AS Return_Date
FROM Issue
WHERE Member_Id = 101;

--13 is something new
-- 13) List the data in the book table with category displayed as D for Database, S
for Science, R for RDBMS and O for others.

```

```

SELECT Book_No, Book_Name, Author_Name, Cost,
       CASE
         WHEN Category='Database' THEN 'D'
         WHEN Category='Science'  THEN 'S'
         WHEN Category='RDBMS'    THEN 'R'
         ELSE 'O'
       END AS Category_Code
FROM Books;

-- 14) List all the members that became members in the year 2006.
SELECT *
FROM Member
WHERE TO_CHAR(Acc_Open_Date, 'YYYY') = 2006;

-- 15) List the Lib_Issue_Id, Issue_Date, Return_Date and No of days Book was
issued.
SELECT Lib_Issue_Id, Issue_Date, Return_Date,
       (Return_Date - Issue_Date) AS Days_Issued
FROM Issue
WHERE Return_Date IS NOT NULL;

-- 16) Find the details of the member of the Library in the order of their joining.
SELECT * FROM Member ORDER BY Acc_Open_Date;

-- 17) Display the count of total no of books issued to Member 101.
SELECT COUNT(*) AS Total_Issued FROM Issue
WHERE Member_Id = 101;

-- 18) Display the total penalty due for all the members.
SELECT SUM(Penalty_Amount) AS Total_Penalty FROM Member;

-- 19) Display the total no of members.
SELECT COUNT(*) AS Total_Members FROM Member;

-- 20) Display the total no of books issued.
SELECT COUNT(*) AS Total_Books_Issued FROM Issue;

-- 21) Display the average membership fees paid by all the members.
SELECT AVG(Fees_Paid) AS Avg_Fees FROM Member;

-- 22) Display no of months between issue date and return date for all issue.
SELECT Lib_Issue_Id,
       MONTHS_BETWEEN(Return_Date, Issue_Date) AS Months_Issued
FROM Issue
WHERE Return_Date IS NOT NULL;

-- 23) Display the length of member's name.
SELECT Member_Id, Member_Name, LENGTH(Member_Name) AS Name_Length FROM Member;

-- 24) Display the first 5 characters of membership_type for all members.
SELECT Member_Id, SUBSTR(Membership_Type,1,5) AS Type_Prefix FROM Member;

-- 25) Display the last day of the month of issue date.
SELECT Lib_Issue_Id, Issue_Date, LAST_DAY(Issue_Date) AS Last_Day_Of_Month
FROM Issue;

```