DataBase project

Rusu Mihnea group 1076

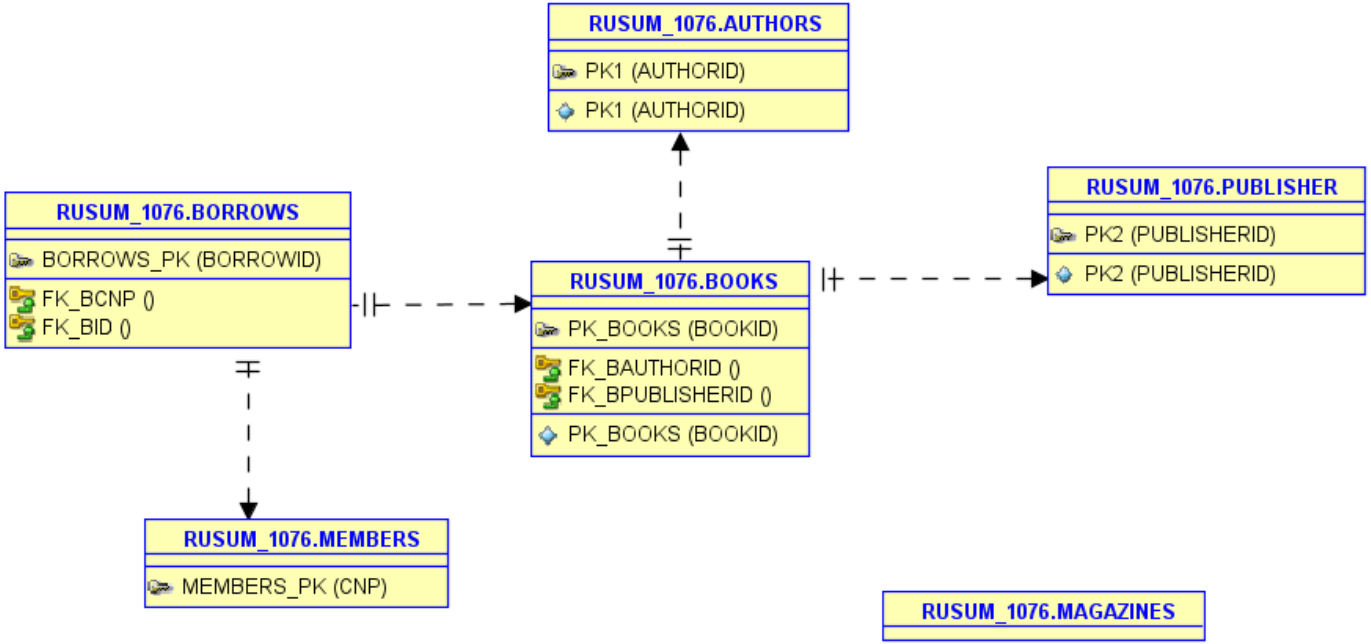
The Library Management System (LMS) project is designed to manage the operations of a modern library. It aims to streamline the processes involved in managing members, books, magazines, authors, publishers, and borrow transactions. The system's primary goal is to provide an easy-to-use interface for administrators and library staff to track the borrowing of resources, manage member information, and maintain an updated collection of books and magazines.

Key Features:

1. **Members Management**: The system will store detailed information about library members, including their name, contact information, and membership details (e.g., joining date, and date of birth). It also allows for tracking of member borrow history.
2. **Books and Magazines Management**: The system manages a collection of books and magazines, storing information such as titles, authors, publishers, and publication details. It allows administrators to add new resources, update existing ones, and remove outdated materials.
3. **Authors and Publishers**: Information related to authors (including their personal details) and publishers (such as contact info) is managed within the system, linking each book to its respective author and publisher.
4. **Borrowing and Returning Resources**: Members can borrow books and magazines, with the system tracking the borrowing and return dates. It also allows for overdue tracking and manages fines based on return dates.
5. **Relational Database Design**: The system uses a relational database where tables are interlinked through foreign keys, ensuring data consistency and integrity. This includes tables like MEMBERS, BOOKS, AUTHORS, PUBLISHER, and BORROWS to model real-world relationships and operations.

The LMS system ensures that all library resources are organized efficiently, helping librarians and staff quickly access the information they need, track the current status of books and magazines, and maintain an up-to-date record of all transactions.

The Database's schema



1.Constructing the database using the DDL statements.

```sql
--Constructing the database using DDL statements
CREATE TABLE MEMBERS(CNP char(12) Primary Key,
membersName varchar(30) not null,
birthDate date,
joinDate date);
CREATE TABLE AUTHORS(
authorID varchar2(10),
authorName varchar2(30),
authorBirthDate date,
Constraint pk1 primary key(authorID)
);
CREATE TABLE PUBLISHER(
publisherID varchar2(10),
publisherName varchar2(30),
phone char(11),
email varchar2(30),
Constraint pk2 primary key(publisherID)
);
```

```sql
CREATE TABLE MAGAZINES (
    magazineID varchar2(10),
    magazineName varchar2(30),
    pagecount number(5),
    issueNumber number(5)
);
CREATE TABLE BOOKS (
    bookID varchar2(10),
    bookName varchar2(100),
    pagecount number(5),
    bauthorID varchar2(10),
    bpublisherID varchar2(10),
    CONSTRAINT fk_bauthorID FOREIGN KEY (bauthorID) REFERENCES AUTHORS(authorID),
    CONSTRAINT fk_bpublisherID FOREIGN KEY (bpublisherID) REFERENCES PUBLISHER(publisherID),
    CONSTRAINT pk_books PRIMARY KEY (bookID)
);
CREATE TABLE BORROWS (
    borrowID VARCHAR2(10) PRIMARY KEY,
    bCNP CHAR(12),
    borrowDate DATE,
    returnDate DATE,
    bID VARCHAR2(10),
    CONSTRAINT fk_bCNP FOREIGN KEY (bCNP) REFERENCES MEMBERS(CNP),
    CONSTRAINT fk_bID FOREIGN KEY (bID) REFERENCES BOOKS(bookID)
);
```

Here are the first three tables.

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CNP	CHAR(12 BYTE)	No	(null)	1	(null)
2	MEMBERSNAME	VARCHAR2(30 BY...	No	(null)	2	(null)
3	BIRTHDATE	DATE	Yes	(null)	3	(null)
4	JOINDATE	DATE	Yes	(null)	4	(null)

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	AUTHORID	VARCHAR2(10 BY...	No	(null)	1	(null)
2	AUTHORNAME	VARCHAR2(30 BY...	Yes	(null)	2	(null)
3	AUTHORBIRTHD...	DATE	Yes	(null)	3	(null)

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PUBLISHERID	VARCHAR2(10 BY...	No	(null)	1	(null)
2	PUBLISHERN...	VARCHAR2(30 BY...	Yes	(null)	2	(null)
3	PHONE	CHAR(11 BYTE)	Yes	(null)	3	(null)
4	EMAIL	VARCHAR2(30 BY...	Yes	(null)	4	(null)

And here’s the following three.

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	BOOKID	VARCHAR2(10 BYTE)	No	(null)	1	(null)
2	BOOKNAME	VARCHAR2(100 BY...	Yes	(null)	2	(null)
3	PAGECOUNT	NUMBER(5,0)	Yes	(null)	3	(null)
4	BAUTHORID	VARCHAR2(10 BYTE)	Yes	(null)	4	(null)
5	BPUBLISHERID	VARCHAR2(10 BYTE)	Yes	(null)	5	(null)

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	BORROWID	VARCHAR2(10 BY...	No	(null)	1	(null)
2	BCNP	CHAR(12 BYTE)	Yes	(null)	2	(null)
3	BORROWDATE	DATE	Yes	(null)	3	(null)
4	RETURNDATE	DATE	Yes	(null)	4	(null)
5	BID	VARCHAR2(10 BY...	Yes	(null)	5	(null)

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	MAGAZINEID	VARCHAR2(10 BY...	Yes	(null)	1	(null)
2	MAGAZINENAME	VARCHAR2(30 BY...	Yes	(null)	2	(null)
3	PAGECOUNT	NUMBER(5,0)	Yes	(null)	3	(null)

I initially added a new field to track the count of issues in magazines, but I later decided to scrap the `MAGAZINES` table altogether

```
ALTER TABLE MAGAZINES ADD issueNumber number(5);
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	MAGAZINEID	VARCHAR2(10 BY...	Yes	(null)	1	(null)
2	MAGAZINENAME	VARCHAR2(30 BY...	Yes	(null)	2	(null)
3	PAGECOUNT	NUMBER(5,0)	Yes	(null)	3	(null)
4	ISSUENUMBER	NUMBER(5,0)	Yes	(null)	4	(null)

```
DROP TABLE MAGAZINES;
```

Script Output | Query Result

Task completed in 0.142 seconds

```
Table MAGAZINES altered.

Table MAGAZINES dropped.
```

2.Using DML statements

The following script demonstrates the insertion of records into the MEMBERS table, along with the resulting data after successful execution.

```oracle
--inserting members
INSERT INTO MEMBERS (CNP, membersName, birthDate, joinDate) VALUES
('183456789012', 'Rusu Mihnea', TO_DATE('2004-05-26', 'YYYY-MM-DD'), TO_DATE('2021-01-09', 'YYYY-MM-DD'));
INSERT INTO MEMBERS (CNP, membersName, birthDate, joinDate) VALUES
('123456789012', 'Rares Popescu', TO_DATE('1990-01-15', 'YYYY-MM-DD'), TO_DATE('2021-01-10', 'YYYY-MM-DD'));
INSERT INTO MEMBERS (CNP, membersName, birthDate, joinDate) VALUES
('234567890123', 'Maria Dragan', TO_DATE('1985-07-23', 'YYYY-MM-DD'), TO_DATE('2022-03-15', 'YYYY-MM-DD'));
INSERT INTO MEMBERS (CNP, membersName, birthDate, joinDate) VALUES
('345678901234', 'Elena Gheorghe', TO_DATE('1992-03-11', 'YYYY-MM-DD'), TO_DATE('2023-05-20', 'YYYY-MM-DD'));
INSERT INTO MEMBERS (CNP, membersName, birthDate, joinDate) VALUES
('456789012345', 'Mihai Dumitrescu', TO_DATE('1988-06-18', 'YYYY-MM-DD'), TO_DATE('2021-09-25', 'YYYY-MM-DD'));
INSERT INTO MEMBERS (CNP, membersName, birthDate, joinDate) VALUES
('567890123456', 'Ana Rusu', TO_DATE('1995-09-30', 'YYYY-MM-DD'), TO_DATE('2022-11-10', 'YYYY-MM-DD'));
INSERT INTO MEMBERS (CNP, membersName, birthDate, joinDate) VALUES
('678901234567', 'Vlad Barbalata', TO_DATE('1980-11-12', 'YYYY-MM-DD'), TO_DATE('2023-06-15', 'YYYY-MM-DD'));
INSERT INTO MEMBERS (CNP, membersName, birthDate, joinDate) VALUES
('789012345678', 'Gabriela Stoica', TO_DATE('1999-04-05', 'YYYY-MM-DD'), TO_DATE('2024-01-05', 'YYYY-MM-DD'));
INSERT INTO MEMBERS (CNP, membersName, birthDate, joinDate) VALUES
('890123456789', 'Constantin Radu', TO_DATE('1975-02-25', 'YYYY-MM-DD'), TO_DATE('2021-08-15', 'YYYY-MM-DD'));
INSERT INTO MEMBERS (CNP, membersName, birthDate, joinDate) VALUES
('901234567890', 'Mihai Druga', TO_DATE('1994-12-13', 'YYYY-MM-DD'), TO_DATE('2022-07-01', 'YYYY-MM-DD'));
INSERT INTO MEMBERS (CNP, membersName, birthDate, joinDate) VALUES
('012345678901', 'Patrick Andrei', TO_DATE('1983-08-21', 'YYYY-MM-DD'), TO_DATE('2024-10-10', 'YYYY-MM-DD'));
```

1	123456789012	Rares Popescu	15-JAN-90	10-JAN-21
2	234567890123	Maria Dragan	23-JUL-85	15-MAR-22
3	345678901234	Elena Gheorghe	11-MAR-92	20-MAY-23
4	456789012345	Mihai Dumitre...	18-JUN-88	25-SEP-21
5	567890123456	Ana Rusu	30-SEP-95	10-NOV-22
6	678901234567	Vlad Barbalata	12-NOV-80	15-JUN-23
7	789012345678	Gabriela Stoica	05-APR-99	05-JAN-24
8	890123456789	Constantin Radu	25-FEB-75	15-AUG-21
9	901234567890	Mihai Druga	13-DEC-94	01-JUL-22
10	012345678901	Patrick Andrei	21-AUG-83	10-OCT-24
11	183456789012	Rusu Mihnea	26-MAY-04	09-JAN-21

The following script demonstrates the insertion of records into the AUTHORS table, along with the resulting data after successful execution.

```sql
-- Inserting authors
INSERT INTO AUTHORS (authorID, authorName, authorBirthDate) VALUES
('AUTH001', 'J.K. Rowling', TO_DATE('1965-07-31', 'YYYY-MM-DD'));
INSERT INTO AUTHORS (authorID, authorName, authorBirthDate) VALUES
('AUTH002', 'George Orwell', TO_DATE('1903-06-25', 'YYYY-MM-DD'));
INSERT INTO AUTHORS (authorID, authorName, authorBirthDate) VALUES
('AUTH003', 'Jane Austen', TO_DATE('1775-12-16', 'YYYY-MM-DD'));
INSERT INTO AUTHORS (authorID, authorName, authorBirthDate) VALUES
('AUTH004', 'William Shakespeare', TO_DATE('1564-04-23', 'YYYY-MM-DD'));
INSERT INTO AUTHORS (authorID, authorName, authorBirthDate) VALUES
('AUTH005', 'Mark Twain', TO_DATE('1835-11-30', 'YYYY-MM-DD'));
INSERT INTO AUTHORS (authorID, authorName, authorBirthDate) VALUES
('AUTH006', 'Haruki Murakami', TO_DATE('1949-01-12', 'YYYY-MM-DD'));
```

	AUTHORID	AUTHORNAME	AUTHORBIRTHDATE
1	AUTH001	J.K. Rowling	31-JUL-65
2	AUTH002	George Orwell	25-JUN-03
3	AUTH003	Jane Austen	16-DEC-75
4	AUTH004	William Shakespe...	23-APR-64
5	AUTH005	Mark Twain	30-NOV-35
6	AUTH006	Haruki Murakami	12-JAN-49

The following script demonstrates the insertion of records into the PUBLISHERS table, along with the resulting data after successful execution.

```sql
--inserting publishers
INSERT INTO PUBLISHER (publisherID, publisherName, phone, email) VALUES
('PUB001', 'Penguin Random House', '12345678901', 'contact@penguinrandomhouse.com');
INSERT INTO PUBLISHER (publisherID, publisherName, phone, email) VALUES
('PUB002', 'HarperCollins', '23456789012', 'info@harpercollins.com');
INSERT INTO PUBLISHER (publisherID, publisherName, phone, email) VALUES
('PUB003', 'Macmillan Publishers', '34567890123', 'support@macmillan.com');
INSERT INTO PUBLISHER (publisherID, publisherName, phone, email) VALUES
('PUB004', 'Editura Humanitas', '40123456789', 'contact@humanitas.ro');
```

	PUBLISHERID	PUBLISHERNAME	PHONE	EMAIL
1	PUB001	Penguin Random Ho...	12345678901	contact@penguinrandomhouse....
2	PUB002	HarperCollins	23456789012	info@harpercollins.com
3	PUB003	Macmillan Publish...	34567890123	support@macmillan.com
4	PUB004	Editura Humanitas	40123456789	contact@humanitas.ro

```sql
INSERT INTO BOOKS (bookID, bookName, pagecount, bauthorID, bpublisherID) VALUES
('BOOK001', 'Harry Potter 1: Philosopher\Stone', 309, 'AUTH001', 'PUB001');
INSERT INTO BOOKS (bookID, bookName, pagecount, bauthorID, bpublisherID) VALUES
('BOOK002', 'Harry Potter 2: Chamber of Secrets', 341, 'AUTH001', 'PUB001');
INSERT INTO BOOKS (bookID, bookName, pagecount, bauthorID, bpublisherID) VALUES
('BOOK003', 'Harry Potter 3: Prisoner of Azkaban', 435, 'AUTH001', 'PUB001');
INSERT INTO BOOKS (bookID, bookName, pagecount, bauthorID, bpublisherID) VALUES
('BOOK004', 'Harry Potter 4: Goblet of Fire', 636, 'AUTH001', 'PUB001');
INSERT INTO BOOKS (bookID, bookName, pagecount, bauthorID, bpublisherID) VALUES
('BOOK005', 'Harry Potter 5: Order of the Phoenix', 766, 'AUTH001', 'PUB001');
INSERT INTO BOOKS (bookID, bookName, pagecount, bauthorID, bpublisherID) VALUES
('BOOK006', 'Harry Potter 6: Half-Blood Prince', 607, 'AUTH001', 'PUB001');
INSERT INTO BOOKS (bookID, bookName, pagecount, bauthorID, bpublisherID) VALUES
('BOOK007', 'Harry Potter 7: Deathly Hallows', 607, 'AUTH001', 'PUB001');
INSERT INTO BOOKS (bookID, bookName, pagecount, bauthorID, bpublisherID) VALUES
('BOOK008', 'Animal Farm', 112, 'AUTH002', 'PUB002');
INSERT INTO BOOKS (bookID, bookName, pagecount, bauthorID, bpublisherID) VALUES
('BOOK009', 'Down and Out in Paris and London', 213, 'AUTH002', 'PUB002');
INSERT INTO BOOKS (bookID, bookName, pagecount, bauthorID, bpublisherID) VALUES
('BOOK010', 'Sense and Sensibility', 409, 'AUTH003', 'PUB003');
INSERT INTO BOOKS (bookID, bookName, pagecount, bauthorID, bpublisherID) VALUES
('BOOK011', 'Emma', 474, 'AUTH003', 'PUB003');
INSERT INTO BOOKS (bookID, bookName, pagecount, bauthorID, bpublisherID) VALUES
('BOOK012', 'Romeo and Juliet', 134, 'AUTH004', 'PUB004');
INSERT INTO BOOKS (bookID, bookName, pagecount, bauthorID, bpublisherID) VALUES
('BOOK013', 'Macbeth', 128, 'AUTH004', 'PUB004');
INSERT INTO BOOKS (bookID, bookName, pagecount, bauthorID, bpublisherID) VALUES
('BOOK014', 'The Adventures of Tom Sawyer', 274, 'AUTH005', 'PUB001');
INSERT INTO BOOKS (bookID, bookName, pagecount, bauthorID, bpublisherID) VALUES
('BOOK015', 'The Prince and the Pauper', 242, 'AUTH005', 'PUB002');
INSERT INTO BOOKS (bookID, bookName, pagecount, bauthorID, bpublisherID) VALUES
('BOOK016', 'Kafka on the Shore', 505, 'AUTH006', 'PUB002');
INSERT INTO BOOKS (bookID, bookName, pagecount, bauthorID, bpublisherID) VALUES
('BOOK017', '1Q84', 928, 'AUTH006', 'PUB003');
```

	BOOKID	BOOKNAME	PAGECOUNT	BAUTHORID	BPUBLISHERID
1	BOOK001	Harry Potter 1: Philosopher\Stone	309	AUTH001	PUB001
2	BOOK002	Harry Potter 2: Chamber of Secrets	341	AUTH001	PUB001
3	BOOK003	Harry Potter 3: Prisoner of Azkaban	435	AUTH001	PUB001
4	BOOK004	Harry Potter 4: Goblet of Fire	636	AUTH001	PUB001
5	BOOK005	Harry Potter 5: Order of the Phoe...	766	AUTH001	PUB001
6	BOOK006	Harry Potter 6: Half-Blood Prince	607	AUTH001	PUB001
7	BOOK007	Harry Potter 7: Deathly Hallows	607	AUTH001	PUB001
8	BOOK008	Animal Farm	112	AUTH002	PUB002
9	BOOK009	Down and Out in Paris and London	213	AUTH002	PUB002
10	BOOK010	Sense and Sensibility	409	AUTH003	PUB003
11	BOOK011	Emma	474	AUTH003	PUB003
12	BOOK012	Romeo and Juliet	134	AUTH004	PUB004
13	BOOK013	Macbeth	128	AUTH004	PUB004
14	BOOK014	The Adventures of Tom Sawyer	274	AUTH005	PUB001
15	BOOK015	The Prince and the Pauper	242	AUTH005	PUB002
16	BOOK016	Kafka on the Shore	505	AUTH006	PUB002
17	BOOK017	1Q84	928	AUTH006	PUB003

The following script demonstrates the insertion of records into the BOOKS table, along with the resulting data after successful execution.

The following script demonstrates the insertion of records into the BORROWS table, along with the resulting data after successful execution.

```sql
INSERT INTO BORROWS (borrowID, bCNP, borrowDate, returnDate, bID) VALUES
('BOR001', '123456789012', TO_DATE('2024-01-10', 'YYYY-MM-DD'), TO_DATE('2024-01-20', 'YYYY-MM-DD'), 'BOOK001');
INSERT INTO BORROWS (borrowID, bCNP, borrowDate, returnDate, bID) VALUES
('BOR002', '234567890123', TO_DATE('2024-01-15', 'YYYY-MM-DD'), TO_DATE('2024-01-25', 'YYYY-MM-DD'), 'BOOK002');
INSERT INTO BORROWS (borrowID, bCNP, borrowDate, returnDate, bID) VALUES
('BOR003', '345678901234', TO_DATE('2024-01-20', 'YYYY-MM-DD'), TO_DATE('2024-01-30', 'YYYY-MM-DD'), 'BOOK003');
INSERT INTO BORROWS (borrowID, bCNP, borrowDate, returnDate, bID) VALUES
('BOR004', '456789012345', TO_DATE('2024-01-25', 'YYYY-MM-DD'), TO_DATE('2024-02-05', 'YYYY-MM-DD'), 'BOOK004');
INSERT INTO BORROWS (borrowID, bCNP, borrowDate, returnDate, bID) VALUES
('BOR005', '567890123456', TO_DATE('2023-12-20', 'YYYY-MM-DD'), TO_DATE('2024-01-01', 'YYYY-MM-DD'), 'BOOK005');

-- Borrowed but not returned yet
INSERT INTO BORROWS (borrowID, bCNP, borrowDate, returnDate, bID) VALUES
('BOR006', '678901234567', TO_DATE('2024-01-05', 'YYYY-MM-DD'), NULL, 'BOOK006');
INSERT INTO BORROWS (borrowID, bCNP, borrowDate, returnDate, bID) VALUES
('BOR007', '789012345678', TO_DATE('2023-12-25', 'YYYY-MM-DD'), NULL, 'BOOK007');
INSERT INTO BORROWS (borrowID, bCNP, borrowDate, returnDate, bID) VALUES
('BOR008', '890123456789', TO_DATE('2024-01-10', 'YYYY-MM-DD'), NULL, 'BOOK008');

-- Mixed borrow-return history
INSERT INTO BORROWS (borrowID, bCNP, borrowDate, returnDate, bID) VALUES
('BOR009', '901234567890', TO_DATE('2024-01-05', 'YYYY-MM-DD'), TO_DATE('2024-01-15', 'YYYY-MM-DD'), 'BOOK009');
INSERT INTO BORROWS (borrowID, bCNP, borrowDate, returnDate, bID) VALUES
('BOR010', '012345678901', TO_DATE('2024-01-08', 'YYYY-MM-DD'), TO_DATE('2024-01-18', 'YYYY-MM-DD'), 'BOOK010');
INSERT INTO BORROWS (borrowID, bCNP, borrowDate, returnDate, bID) VALUES
('BOR011', '123456789012', TO_DATE('2024-01-12', 'YYYY-MM-DD'), TO_DATE('2024-01-22', 'YYYY-MM-DD'), 'BOOK011');
INSERT INTO BORROWS (borrowID, bCNP, borrowDate, returnDate, bID) VALUES
('BOR012', '234567890123', TO_DATE('2024-01-15', 'YYYY-MM-DD'), NULL, 'BOOK012');
```

	BORROWID	BCNP	BORROWDATE	RETURNDATE	BID
1	BOR001	123456789012	10-JAN-24	20-JAN-24	BOOK001
2	BOR002	234567890123	15-JAN-24	25-JAN-24	BOOK002
3	BOR003	345678901234	20-JAN-24	30-JAN-24	BOOK003
4	BOR004	456789012345	25-JAN-24	05-FEB-24	BOOK004
5	BOR005	567890123456	20-DEC-23	01-JAN-24	BOOK005
6	BOR006	678901234567	05-JAN-24	(null)	BOOK006
7	BOR007	789012345678	25-DEC-23	(null)	BOOK007
8	BOR008	890123456789	10-JAN-24	(null)	BOOK008
9	BOR009	901234567890	05-JAN-24	15-JAN-24	BOOK009
10	BOR010	012345678901	08-JAN-24	18-JAN-24	BOOK010
11	BOR011	123456789012	12-JAN-24	22-JAN-24	BOOK011
12	BOR012	234567890123	15-JAN-24	(null)	BOOK012

A member wants their data deleted,we do this by running the following sequence:

```sql
DELETE FROM BORROWS
WHERE bCNP = (SELECT CNP FROM MEMBERS WHERE membersName = 'Elena Gheorghe');
DELETE FROM MEMBERS
WHERE membersName = 'Elena Gheorghe';
```

The output and updated table:

Script Output | Query Result

Task completed in 0.043 seconds

```
1 row deleted.

1 row deleted.
```

	CNP	MEMBERSNAME	BIRTHDATE	JOINDATE
1	183456789012	Rusu Mihnea	26-MAY-04	09-JAN-21
2	123456789012	Rares Popescu	15-JAN-90	10-JAN-21
3	234567890123	Maria Dragan	23-JUL-85	15-MAR-22
4	456789012345	Mihai Dumitre...	18-JUN-88	25-SEP-21
5	567890123456	Ana Rusu	30-SEP-95	10-NOV-22
6	678901234567	Vlad Barbalata	12-NOV-80	15-JUN-23
7	789012345678	Gabriela Stoica	05-APR-99	05-JAN-24
8	890123456789	Constantin Radu	25-FEB-75	15-AUG-21
9	901234567890	Mihai Druga	13-DEC-94	01-JUL-22
10	012345678901	Patrick Andrei	21-AUG-83	10-OCT-24

A member got married and wanted to change their name in the database to match their new one.

	CNP	MEMBERSNAME	BIRTHDATE	JOINDATE
1	183456789012	Rusu Mihnea	26-MAY-04	09-JAN-21
2	123456789012	Rares Popescu	15-JAN-90	10-JAN-21
3	234567890123	Maria Dragan	23-JUL-85	15-MAR-22
4	456789012345	Mihai Dumitre...	18-JUN-88	25-SEP-21
5	567890123456	Ana Draghici	30-SEP-95	10-NOV-22
6	678901234567	Vlad Barbalata	12-NOV-80	15-JUN-23
7	789012345678	Gabriela Stoica	05-APR-99	05-JAN-24
8	890123456789	Constantin Radu	25-FEB-75	15-AUG-21
9	901234567890	Mihai Druga	13-DEC-94	01-JUL-22
10	012345678901	Patrick Andrei	21-AUG-83	10-OCT-24

```
UPDATE MEMBERS
SET membersName = 'Ana Draghici'
WHERE membersName = 'Ana Rusu';
```

Script Output | Query Result

Task completed in 0.029 sec

```
1 row updated.
```

3.Diverse and relevant SELECT statements for the project theme

Retrieves names and join dates of members who joined after January 1, 2022, but before today.

```sql
SELECT membersName, joinDate
FROM MEMBERS
WHERE joinDate >= TO_DATE('2022-01-01', 'YYYY-MM-DD')
  AND joinDate < SYSDATE;
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 6 in 0.008 seconds

	MEMBERSNAME	JOINDATE
1	Maria Dragan	15-MAR-22
2	Ana Draghici	10-NOV-22
3	Vlad Barbalata	15-JUN-23
4	Gabriela Sto...	05-JAN-24
5	Mihai Druga	01-JUL-22
6	Patrick Andrei	10-OCT-24

Retrieves books with page counts bigger than 300 and a non-null publisher.

```
185   SELECT bookName, pagecount FROM BOOKS WHERE pagecount >300  AND bpublisherID IS NOT NULL;
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 11 in 0.009 seconds

	BOOKNAME	PAGECOUNT
1	Harry Potter 1: Philosopher\Stone	309
2	Harry Potter 2: Chamber of Secrets	341
3	Harry Potter 3: Prisoner of Azkaban	435
4	Harry Potter 4: Goblet of Fire	636
5	Harry Potter 5: Order of the Phoe...	766
6	Harry Potter 6: Half-Blood Prince	607
7	Harry Potter 7: Deathly Hallows	607
8	Sense and Sensibility	409
9	Emma	474
10	Kafka on the Shore	505
11	1Q84	928

Retrieves the author and book name of all books that contain the word “Harry”.

```sql
SELECT B.bookName, A.authorName
FROM BOOKS B
JOIN AUTHORS A ON B.bauthorID = A.authorID
WHERE B.bookName LIKE '%Harry%';
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 7 in 0.009 seconds

	BOOKNAME	AUTHORNAME
1	Harry Potter 1: Philosopher\Stone	J.K. Rowling
2	Harry Potter 2: Chamber of Secrets	J.K. Rowling
3	Harry Potter 3: Prisoner of Azkaban	J.K. Rowling
4	Harry Potter 4: Goblet of Fire	J.K. Rowling
5	Harry Potter 5: Order of the Phoe...	J.K. Rowling
6	Harry Potter 6: Half-Blood Prince	J.K. Rowling
7	Harry Potter 7: Deathly Hallows	J.K. Rowling

This query will return the names and birthdates of members whose birth year is either 1985, 1992, or 1995:

```
SELECT membersName, birthDate FROM MEMBERS WHERE EXTRACT(YEAR FROM birthDate) IN (1985, 1992, 1995)
```

Script Output | Query Result

SQL | All Rows Fetched:

	MEMBERSNAME	BIRTHDATE
1	Maria Dragan	23-JUL-85
2	Ana Draghici	30-SEP-95

This query will return the name and number of pages of books that have between 100 and 250 pages:

```
SELECT bookName, pagecount FROM BOOKS WHERE pagecount BETWEEN 100 AND 250;
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 5 in 0.011 seconds

	BOOKNAME	PAGECOUNT
1	Animal Farm	112
2	Down and Out in Paris and Lon...	213
3	Romeo and Juliet	134
4	Macbeth	128
5	The Prince and the Pauper	242

```sql
SELECT A.authorName, B.bauthorID, COUNT(*) AS books_count, AVG(B.pagecount) AS avg_pages
FROM BOOKS B
JOIN AUTHORS A ON B.bauthorID = A.authorID
GROUP BY A.authorName, B.bauthorID
HAVING COUNT(*) > 1;
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 6 in 0.014 seconds

	AUTHORNAME	BAUTHORID	BOOKS_COUNT	AVG_PAGES
1	J.K. Rowling	AUTH001	7	28.7142857142857142857142857142857 14...
2	George Orwell	AUTH002	2	162.5
3	Jane Austen	AUTH003	2	441.5
4	William Shakespe...	AUTH004	2	131
5	Mark Twain	AUTH005	2	258
6	Haruki Murakami	AUTH006	2	716.5

This query will return the author's name, the number of books they have written, and the average page count of their books for authors who have written more than one book.

The query finds and returns the names of members who have at least one book that they have borrowed but have not yet returned.

```
205   SELECT M.membersName FROM MEMBERS M WHERE EXISTS (SELECT 1 FROM BORROWS BR WHERE BR.bCNP = M.CNP AND BR.returnDate IS NULL);
```

Script Output x | Query Re

SQL | All Row

	MEMBERSNAME
1	Maria Dragan
2	Vlad Barbalata
3	Gabriela Sto...
4	Constantin R...

```
206   SELECT authorName AS name FROM AUTHORS
207   UNION SELECT publisherName AS name FROM PUBLISHER;
```

	NAME
1	J.K. Rowling
2	George Orwell
3	Jane Austen
4	William Shakespeare
5	Mark Twain
6	Haruki Murakami
7	Penguin Random House
8	HarperCollins
9	Macmillan Publishers
10	Editura Humanitas

The query displays all the authors names and all the publisher names.

```
210 SELECT membersName FROM MEMBERS INTERSECT SELECT M.membersName FROM MEMBERS M INNER JOIN BORROWS BR ON M.CNP = BR.bCNP;
```

Script Output | Query Result | Query

SQL | All Rows Fetched: 9 in 0.0

	MEMBERSNAME
1	Rares Popescu
2	Maria Dragan
3	Mihai Dumitre...
4	Ana Draghici
5	Vlad Barbalata
6	Gabriela Stoica
7	Constantin Radu
8	Mihai Druga
9	Patrick Andrei

The query displays all the names of the members who have borrowed a book.

```sql
SELECT membersName, birthDate,
  CASE
    WHEN EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM birthDate) > 40 THEN 'Adult'
    WHEN EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM birthDate) BETWEEN 18 AND 40 THEN 'Young-Adult'
    ELSE 'Minor'
  END AS age_group
FROM MEMBERS;
```

This query dynamically categorizes members into three age groups: Adult, Young-Adult, and Minor based on their birthdate and the current date.

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 10 in 0.014 seconds

	MEMBERSNAME	BIRTHDATE	AGE_GROUP
1	Rusu Mihnea	26-MAY-04	Young-Adult
2	Rares Popescu	15-JAN-90	Young-Adult
3	Maria Dragan	23-JUL-85	Young-Adult
4	Mihai Dumitre...	18-JUN-88	Young-Adult
5	Ana Draghici	30-SEP-95	Young-Adult
6	Vlad Barbalata	12-NOV-80	Adult
7	Gabriela Stoica	05-APR-99	Young-Adult
8	Constantin Radu	25-FEB-75	Adult
9	Mihai Druga	13-DEC-94	Young-Adult
10	Patrick Andrei	21-AUG-83	Adult

This query creates a view of the members who are currently borrowing a book.

```
CREATE VIEW ActiveBorrowers AS
SELECT M.membersName, B.bookName
FROM MEMBERS M
JOIN BORROWS BR ON M.CNP = BR.bCNP
JOIN BOOKS B ON BR.bID = B.bookID
WHERE BR.returnDate IS NULL;
```

Script Output | Query Result | Query R

Task completed in 0.22 secc

```
View ACTIVEBORROWERS created.
```

	MEMBERSNAME	BOOKNAME
1	Vlad Barbalata	Harry Potter 6: Half-Blood Pri...
2	Gabriela Sto...	Harry Potter 7: Deathly Hallows
3	Constantin R...	Animal Farm
4	Maria Dragan	Romeo and Juliet

Find the member with the most books borrowed.

```sql
SELECT m.membersName, COUNT(*) AS borrow_count
FROM MEMBERS m
JOIN BORROWS br ON m.CNP = br.bCNP
GROUP BY m.membersName
ORDER BY borrow_count DESC;
```

	MEMBERSNAME	BORROW_COUNT
1	Maria Dragan	2
2	Rares Popescu	2
3	Patrick Andrei	1
4	Gabriela Stoica	1
5	Constantin Radu	1
6	Mihai Druga	1
7	Ana Draghici	1
8	Mihai Dumitre...	1
9	Vlad Barbalata	1

Find publishers with at least 3 books in the library:

```sql
SELECT p.publisherName, COUNT(b.bookID) AS book_count
FROM PUBLISHER p
JOIN BOOKS b ON p.publisherID = b.bpublisherID
GROUP BY p.publisherName
HAVING COUNT(b.bookID) >= 3;
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 3 in 0.013 seconds

	PUBLISHERNAME	BOOK_COUNT
1	Penguin Random Ho...	8
2	HarperCollins	4
3	Macmillan Publish...	3

Find books that have never been borrowed:

```sql
SELECT b.bookID, b.bookName
FROM BOOKS b
LEFT JOIN BORROWS br ON b.bookID = br.bID
WHERE br.borrowID IS NULL;
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 6 in 0.012 seconds

	BOOKID	BOOKNAME
1	BOOK003	Harry Potter 3: Prisoner of Azka...
2	BOOK013	Macbeth
3	BOOK014	The Adventures of Tom Sawyer
4	BOOK015	The Prince and the Pauper
5	BOOK016	Kafka on the Shore
6	BOOK017	1Q84

Find the average page count of books published by 'Penguin Random House':

```sql
SELECT AVG(b.pagecount) AS avg_page_count
FROM BOOKS b
JOIN PUBLISHER p ON b.bpublisherID = p.publisherID
WHERE p.publisherName = 'Penguin Random House';
```

Script Output | Query R

SQL | All Ro

	AVG_PAGE_COUNT
1	496.875

Thank you!