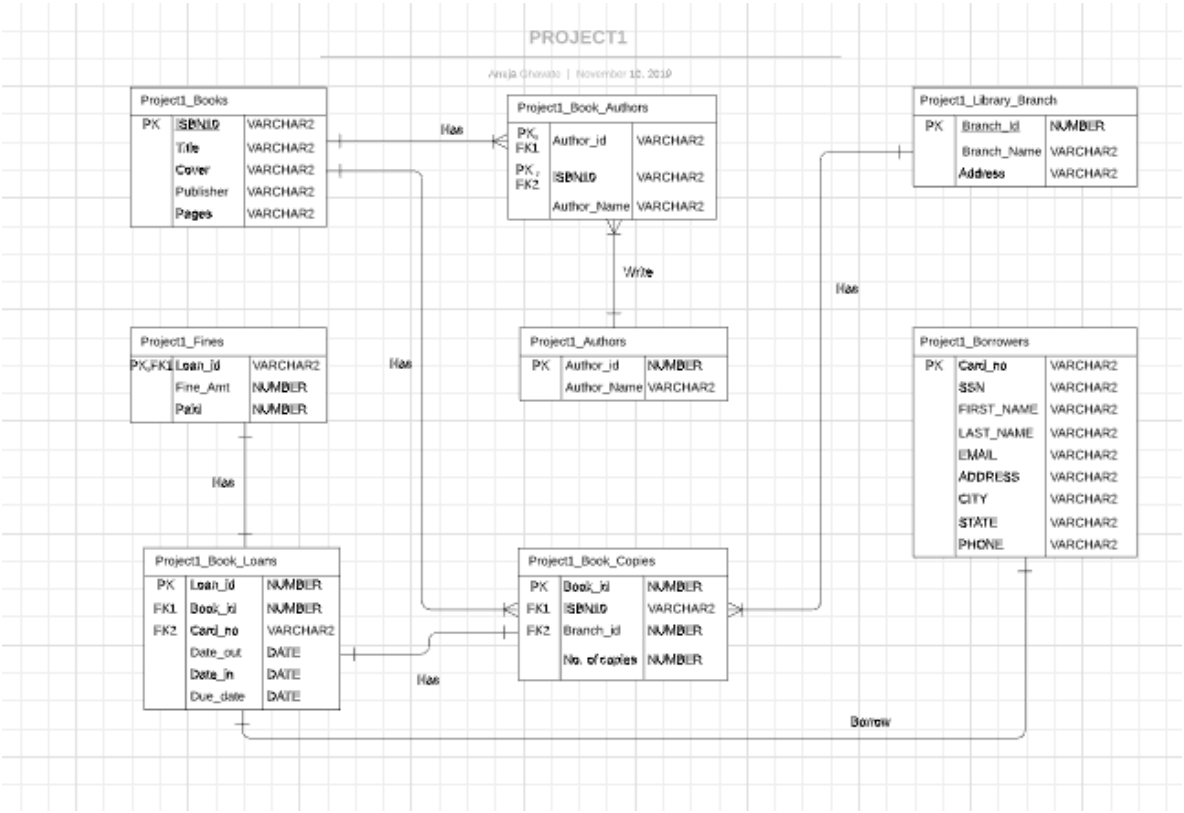
Entity Relationship Diagram



CODE TO CREATE TABLE PROJECT1_BOOK_COPIES_LOAD:

```
CREATE TABLE project1_book_copies(
book_id NUMBER(38,0),
ISBN10 VARCHAR2(100) REFERENCES project1_books(ISBN10),
Branch_id NUMBER(10,0) REFERENCES project1_library_Branch(branch_id),
No_of_copies NUMBER(10,0),
PRIMARY KEY(book_id));
```

CODE TO CREATE SEQUENCE FOR project1_book_copies TABLE TO GENERATE book_id:

```
CREATE SEQUENCE primary_seq
START WITH 1
INCREMENT BY 1;
```

CODE TO CREATE TABLE PROJECT1_BOOKS_LOAD:

```
CREATE TABLE project1_books_load
(ISBN10 VARCHAR2 (100),
 ISBN13 VARCHAR2 (100),
 Title VARCHAR2(1000),
 Authro VARCHAR2(100),
 Cover VARCHAR2 (100),
 Publisher VARCHAR2(100),
 Pages NUMBER(10,0),
CONSTRAINT BooksPK PRIMARY KEY (ISBN10));
```

CODE TO CREATE VIEW Authors_ISBN :

```
CREATE VIEW Authors_ISBN AS
SELECT authro,ISBN10
,REGEXP_SUBSTR (authro, '[^,]+', 1, 1) AS part_1
, REGEXP_SUBSTR (authro, '[^,]+', 1, 2) AS part_2
, REGEXP_SUBSTR (authro, '[^,]+', 1, 3) AS part_3
, REGEXP_SUBSTR (authro, '[^,]+', 1, 4) AS part_4
, REGEXP_SUBSTR (authro, '[^,]+', 1, 5) AS part_5
FROM project1_books_load WHERE REGEXP_SUBSTR (authro, '[^,]+', 1, 1) IS NOT
NULL;
```

CODE TO INSERT DATA INTO project1_authors TABLE FROM Authors_ISBN VIEW:

```
INSERT INTO project1_authors(AUTHOR_NAME)
(SELECT DISTINCT (part_1)
FROM (
SELECT part_1 FROM Authors_ISBN UNION
SELECT part_2 FROM Authors_ISBN  UNION
SELECT part_3 FROM Authors_ISBN UNION
SELECT part_4 FROM Authors_ISBN UNION
SELECT part_5 FROM Authors_ISBN)
WHERE part_1 IS NOT NULL);
```

CODE TO CREATE project1_library_branch_load TABLE:

```
CREATE TABLE project1_library_branch_load
(Branch_id NUMBER(10,0),
 Branch_name VARCHAR2(100),
 address VARCHAR2(100),
CONSTRAINT library_branchPK PRIMARY KEY (Branch_id));
```

CODE TO CREATE TEMPORARY TABLE temp_auth :

```
CREATE TABLE temp_auth (
auth_name VARCHAR2(30 BYTE));
```

CODE TO CREATE Project1_authors TABLES:

```
CREATE TABLE Project1_authors (
author_id number(38,0),
author_name varchar2(300 byte),
PRIMARY KEY(author_id));
```

CODE TO CREATE SEQUENCE Project1_authors TO GENERATE author_id :

```
CREATE SEQUENCE Project1_authors
INCREMENT BY 1
START WITH 1
NOCYCLE
CACHE 100;
```

CODE TO CREATE project1_book_authors TABLE:

```
CREATE TABLE project1_book_authors
(author_id NUMBER(38,0),
 author_name VARCHAR2(300 BYTE),
 ISBN10 VARCHAR2(100 BYTE),
CONSTRAINT book_authorsPK PRIMARY KEY (author_id, ISBN10),
CONSTRAINT book_authors_Author_idFk FOREIGN KEY (author_id) REFERENCES
project1_authors,
CONSTRAINT book_authors_ISBN10Fk FOREIGN KEY (ISBN10) REFERENCES
project1_books_load);
```

CODE TO CREATE TEMPORARY TABLE project1_books_author_temp:

```
CREATE TABLE project1_books_author_temp(
author_name VARCHAR2(300 BYTE),
ISBN10 VARCHAR2(100 BYTE));
```

CODE TO CREATE FINAL TABLE project1_book_authors:

```
INSERT INTO project1_book_authors
VALUE(SELECT auth.author_id ,  auth.author_name, auth1.isbn10
FROM project1_authors auth INNER JOIN project1_books_author_temp auth1 ON
auth.author_name = auth1.author_name);
```

CODE TO CREATE FINAL TABLE project1_books :

```
CREATE TABLE project1_books (
ISBN10 VARCHAR2(100 BYTE) NOT NULL,
Title VARCHAR2(500 BYTE),
Cover VARCHAR2(100 BYTE),
Publisher VARCHAR2(100 BYTE),
PAGES VARCHAR2(100 BYTE),
PRIMARY KEY(ISBN10));
```

CODE TO INSERT DATA INTO FINAL TABLE project1_books FROM project1_books_load TABLE:

```
INSERT INTO project1_books VALUE
(SELECT ISBN10,Title,Cover,Publisher, PAGES FROM project1_books_load);
```

CODE TO CREATE FINAL TABLE project1_library_Branch:

```
CREATE TABLE project1_library_Branch(
Branch_id NUMBER(38,0),
Branch_name VARCHAR2(100 BYTE),
Address VARCHAR2(100 BYTE),
PRIMARY KEY(branch_id));
```

CODE TO INSERT DATA INTO FINAL TABLE project1_library_Branch FROM project1_library_branch_load TABLE:

```
INSERT INTO project1_library_Branch VALUE
(SELECT Branch_id, Branch_name, Address FROM project1_library_branch_load);
```

CODE TO CREATE FINAL TABLE project1_book_copies:

```
CREATE TABLE project1_book_copies(
book_id NUMBER(38,0),
ISBN10 VARCHAR2(100) REFERENCES project1_books(ISBN10),
Branch_id NUMBER(10,0) REFERENCES project1_library_Branch(branch_id),
No_of_copies NUMBER(10,0),
PRIMARY KEY(book_id));
```

CODE TO INSERT DATA INTO FINAL TABLE project1_book_copies FROM project1_book_copies_load TABLE:

```
INSERT INTO project1_book_copies VALUE( SELECT
book_id,ISBN10,Branch_id,No_of_copies FROM project1_book_copies_load);
```

CODE TO SEPARATE 2 NUMBER OF COPIES INTO 2 DIFFERENT COPIES:

```
INSERT INTO project1_book_copies value(select primary_seq.nextval,
ISBN10,Branch_id,'1' from project1_book_copies where no_of_copies=2);
UPDATE project1_book_copies cop SET no_of_copies=1 WHERE EXISTS( SELECT 1 FROM
project1_book_copies cop1 WHERE cop1.Book_id=cop.Book_id AND no_of_copies=2);
```

CODE TO CREATE TABLE project1_borrowers_load:

```
CREATE TABLE project1_borrowers_load
(Card_no VARCHAR2(40),
 ssn VARCHAR2(40),
 first_name VARCHAR2(20),
 last_name VARCHAR2(20),
 email VARCHAR2(40),
 address VARCHAR2(100),
 city VARCHAR2(20),
 State VARCHAR2(20),
 Phone VARCHAR2(20),
CONSTRAINT BorrowersPK PRIMARY KEY (Card_no));
```

CODE TO CREATE FINAL TABLE project1_borrowers:

```
CREATE TABLE project1_borrowers(
card_no VARCHAR2(20),
ssn VARCHAR2(20),
first_name VARCHAR2(20),
last_name VARCHAR2(20),
email VARCHAR2(40),
address VARCHAR2(50),
city VARCHAR2(20),
state VARCHAR2(20),
phone VARCHAR2(20),
PRIMARY KEY(card_no));
```

CODE TO INSERT DATA INTO FINAL TABLE project1_borrowers FROM project1_borrowers_load TABLE:

```
INSERT INTO project1_borrowers
(SELECT * FROM project1_borrowers_load);
```

CODE TO CREATE FINAL TABLE project1_book_loans:

```
CREATE TABLE project1_book_loans
(Loan_id NUMBER(38,0),
 Book_id NUMBER(38,0),
 Card_no VARCHAR2(40 BYTES),
 Date_out DATE,
 Due_date DATE,
 Date_in DATE,
CONSTRAINT book_loansPK PRIMARY KEY (Loan_id),
CONSTRAINT book_authors_Book_idFk FOREIGN KEY (Book_id) REFERENCES
project1_book_copies,
CONSTRAINT book_authors_Card_noFk FOREIGN KEY (Card_no) REFERENCES
project1_borrowers);
```

CODE TO CREATE SEQUENCE Project1_Loans FOR project1_book_loans TO GENERATE Loan_id:

```
CREATE SEQUENCE Project1_Loans
INCREMENT BY 1
START WITH 1
NOCYCLE
CACHE 100;
```

CODE TO CREATE TEMPORARY TABLES TO JOIN book_id FROM project1_book_copies AND card_no FROM project1_borrowers:

```
CREATE TABLE Temp_card
(temp_no NUMBER(20,0) GENERATED ALWAYS AS IDENTITY NOT NULL,
card_no VARCHAR2(50),
CONSTRAINT Temp_cardpk PRIMARY KEY (card_no)
);

INSERT INTO Temp_card(card_no)
SELECT card_no FROM project1_borrowers;

COMMIT;

CREATE TABLE Temp_book
(temp_id NUMBER(20,0) GENERATED ALWAYS AS IDENTITY NOT NULL,
book_id NUMBER(35,0),
CONSTRAINT Temp_bookpk PRIMARY KEY (book_id)
);

INSERT INTO Temp_book(book_id)
SELECT book_id FROM project1_book_copies;

COMMIT;
```

CODE TO INSERT card_no, book_id, DATE_OUT, DATE_IN ,due_date IN project1_book_loans TABLE:

```
insert into project1_book_loans(card_no,book_id,Date_out,Due_date)
select c.card_no,b.book_id,sysdate,sysdate+15
from Temp_card c,  Temp_book b
where  c.temp_no = b.temp_id;
commit;


 UPDATE project1_Book_loans l
 SET Date_in = sysdate+dbms_random.value(1,30)
 WHERE exists(select 1 from project1_Book_Loans l1 where l1.loan_id=l.loan_id)
and rownum<=1000;
 ROLLBACK;
commit;
```

CODE TO CREATE FINAL TABLE project1_fines:

```
CREATE TABLE project1_fines
(Loan_id VARCHAR2 (40),
 Fine_amt NUMBER(20,0),
 Paid NUMBER(5),
CONSTRAINT Loan_idPK PRIMARY KEY (Loan_id));
```

CREATE TEMPORARY TABLE Temp_Fines:

```
CREATE TABLE Temp_Fines
(Loan_id VARCHAR2(40),
due_date date,
date_in date,
fine_days varchar(40),
CONSTRAINT Temp_Loan_id primary key (Loan_id));
```

INSERT DATA INTO project1_fines: (Note: Calculated fine amount based on $2/day rate)

```
INSERT INTO Temp_Fines(Loan_id,due_date,date_in,fine_days)
SELECT Loan_id,due_date,date_in,round(due_date - date_in ,0) FROM
project1_book_loans;
commit;

INSERT INTO project1_fines(loan_id,Fine_amt,Paid)
SELECT loan_id,fine_days * 2,dbms_random.value(0,1) FROM Temp_Fines WHERE
fine_days>0;
```

REPORT 1: CODE TO FETCH Top 10 most popular books in the last month ON REAL TIME ENTRY

```
SELECT b.title,COUNT(b.isbn10)AS MOST_BORROWED
FROM project1_books b
INNER JOIN project1_book_copies c
ON b.isbn10=c.isbn10
INNER JOIN project1_book_loans l
ON c.book_id=l.book_id
WHERE l.date_out BETWEEN TO_DATE('&DATE1','DD-MM-YY') and TO_DATE('&DATE2','DD-
MM-YY')
GROUP BY b.title
ORDER BY most_borrowed DESC
FETCH FIRST 10 ROW ONLY;
```

REPORT 2: CODE TO FETCH Top 10 books that were checked in late ON REAL TIME ENTRY

```
SELECT bk.title,COUNT(bk.isbn10)AS NO_OF_LATE_CHECK_IN
FROM project1_books bk
INNER JOIN project1_book_copies cp
ON bk.isbn10=cp.isbn10
INNER JOIN project1_book_loans ln
ON cp.book_id=ln.book_id
WHERE ln.date_in>ln.due_date and cp.branch_id=:BranchID
GROUP BY bk.title
ORDER BY no_of_late_check_in DESC
FETCH FIRST 10 ROW ONLY;
```

SEARCH QUERY:

```
SELECT ISBN10,TITLE,AUTHOR_NAME,branch_id FROM
(SELECT bk.ISBN10,bk.Title,auth.author_name,bkcp.branch_id
FROM Project1_books bk
INNER JOIN project1_book_authors auth ON bk.ISBN10=auth.ISBN10
INNER JOIN Project1_book_copies bkcp ON bkcp.isbn10=bk.isbn10
WHERE UPPER(auth.author_name) LIKE UPPER(CONCAT('%',CONCAT('&Author_Name','%')))
AND UPPER(bk.Title) LIKE UPPER(CONCAT('%',CONCAT('&TITLE','%')))
AND bk.ISBN10 LIKE CONCAT('%',CONCAT('&ISBN10','%'))
GROUP BY bk.ISBN10,bk.Title,auth.author_name,bkcp.branch_id);
```