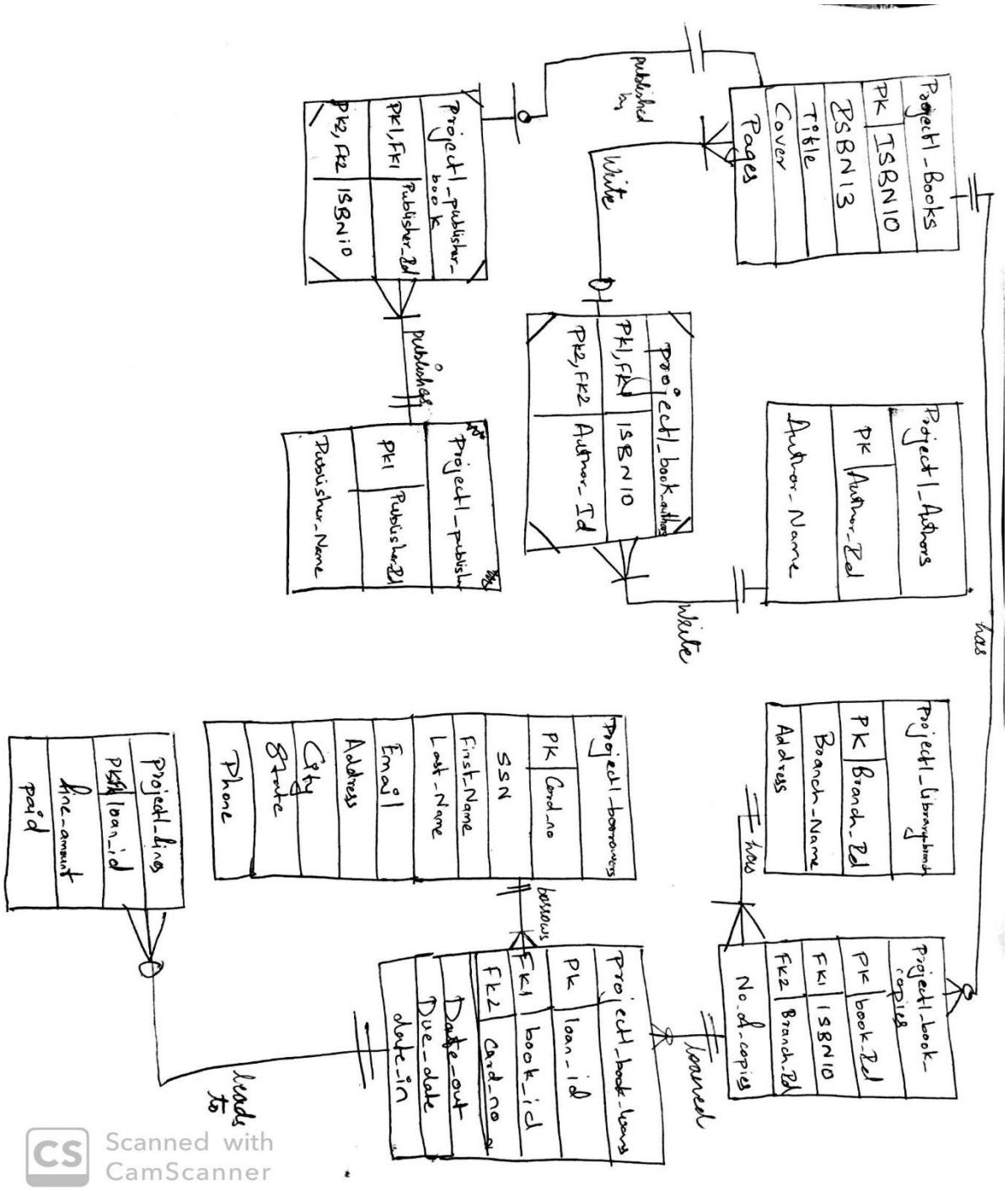
Sidharth Jahagirdar
SSJ180009
Project 1

ERD



Scanned with CamScanner

Create table for initial load files:

Create statement for project1_books_load:

```
create table project1_books_load (ISBN10 VARCHAR2(10), ISBN13
varchar2(13),  Title varchar2(300),  Author varchar2(300),
Cover varchar2(300),  Publisher varchar2(150), Pages number(6));
```

Create statement for project1_borrowers_load:

```
create table project1_borrowers_load (ID0000id VARCHAR2(8),ssn
VARCHAR2(12),first_name VARCHAR2(50),last_name VARCHAR2(50),
email VARCHAR2(200),address VARCHAR2(400),city VARCHAR2(150),state
VARCHAR2(5),phone VARCHAR2(14));
```

Create statement for project1_book_copies_load:

```
create table project1_book_copies_load( book_id varchar(10), branch_id
number(2), no_of_copies number(2));
```

Create statement for project1_library_branch_load:

```
create table project1_library_branch_load(branch_id number(2),
branch_name varchar2(50),address varchar2(400));
```

Create table Queries:

Create table for project1_books:

```
create table project1_books(ISBN10 VARCHAR2(10), ISBN13 varchar2(13),
Title varchar2(300), Cover varchar2(300), Pages number(6));
```

Create table for project1_authors:

```
create table project1_authors(Author_Id integer GENERATED BY DEFAULT AS
IDENTITY (START WITH 1) NOT NULL PRIMARY KEY, Author_Name
varchar2(100));
```

Create table for project1_book_authors:

```
create table project1_book_authors(ISBN10 VARCHAR2(10), Author_Id
NUMBER);
```

Create table for project1_publisher:

```
create table project1_publisher(publisher_id integer GENERATED BY
DEFAULT AS IDENTITY (START WITH 1) NOT NULL PRIMARY KEY,publisher_name
varchar2(300));
```

Create table for project1_book_publisher:

```
create table project1_book_publisher (publisher_id number, ISBN10
VARCHAR2(10));
```

Create table for project1_library_branch:

```
create table project1_library_branch(branch_id number(2),branch_name
varchar2(80),address varchar2(300));
```

Create table for project1_book_copies:

```
create table project1_book_copies( book_id integer GENERATED BY DEFAULT
AS IDENTITY (START WITH 1) NOT NULL PRIMARY KEY,isbn varchar2(10),
branch_id number(2), no_of_copies number(2));
```

Create table for project1_borrowers:

```
create table project1_borrowers(CardNo VARCHAR2(8),ssn
VARCHAR2(12),first_name VARCHAR2(80),last_name VARCHAR2(80),email
VARCHAR2(100),
address VARCHAR2(200),city VARCHAR2(100),state VARCHAR2(2),phone
VARCHAR2(14));
```

Create table for project1_book_loans:

```
create table project1_book_loans(loan_id integer GENERATED BY DEFAULT
AS IDENTITY (START WITH 1) NOT NULL PRIMARY KEY,
book_id integer, cardno varchar2(8),date_out date, due_date date,
date_in date );
```

Create table for project1_fines:

```
create table project1_fines(loan_Id Number, fine_amt NUMBER(5,2), paid
NUMBER(1));
```

CONSTRAINTS:

```
alter table project1_books add constraint Project1_books_PK Primary key
(ISBN10);

alter table project1_book_authors add constraint Book_authors_pk
Primary key(isbn10,author_id);

alter table project1_book_authors add constraint Book_fk foreign
key(isbn10) references project1_books(isbn10);

alter table project1_book_authors add constraint author_fk foreign
key(author_id) references project1_authors(author_id);

alter table project1_book_publisher add constraint Book_publisher_pk
Primary key(publisher_id,isbn10);

alter table project1_book_publisher add constraint Book_Publi_fk
foreign key(isbn10) references project1_books(isbn10);

alter table project1_book_publisher add constraint publisher_fk foreign
key(publisher_id) references project1_publisher(publisher_id);

alter table project1_library_branch add constraint Lib_btranch_pk
Primary key(Branch_id);

alter table project1_borrowers add constraint borrower_pk Primary
key(cardno);

alter table project1_fines add constraint fine_pk Primary key(loan_id);

alter table project1_fines add constraint fine_loan_fk foreign
key(loan_id) references project1_book_loans(loan_id);

alter table project1_book_loans add constraint BookIdFk foreign
key(book_id) references project1_book_copies(book_id);


alter table project1_book_loans add constraint BorrowerFk foreign
key(cardno) references project1_borrowers(cardno);


alter table project1_book_copies add constraint BookISBN_FK foreign
key(isbn) references project1_books(isbn10);


alter table project1_book_copies add constraint Library_Branch_FK foreign
key(branch_id) references project1_library_branch(branch_id);
```

Insert Statements:

project1_books

```
insert into project1_books (isbn10,isbn13,title,cover,pages)(select
isbn10,isbn13,title,cover,pages from project1_books_load);
```

project1_authors

```
select distinct author, fname,lname,mname, trim(trailing ',' from
substr( temp3, 1,instr( temp3, ','  ) )) as pname, trim(leading ','
from substr( temp3,instr( temp3, ','  )) )as tname
from(select distinct isbn10,author, fname,lname, trim(trailing ',' from
substr( temp2, 1,instr( temp2, ','  ) )) as mname, trim(leading ','
from substr( temp2,instr( temp2, ','  )) ) as temp3
from (
select distinct isbn10,author, fname, trim(trailing ',' from substr(
temp, 1,instr( temp, ','  ) )) as lname, trim(leading ',' from substr(
temp,instr( temp, ','  )) )as temp2
from (select distinct isbn10,author, trim(trailing ',' from substr(
author, 1,instr( author, ','  ) )) as fname, trim(leading ',' from
substr( author,instr( author, ','  )) )
as temp from project1_books_load)));
```

note:
used the above output to load data into a temp table column wise and then mapped it with author_id in the author table

project1_book_authors:

```
insert into project1_book_authors(author_id,isbn10)(select
au.author_id, tab1.isbn10 from project1_authors au full outer join
(select distinct isbn10,author, fname,lname,mname, trim(trailing ','
from substr( temp3, 1,instr( temp3, ','  ) )) as pname, trim(leading
',' from substr( temp3,instr( temp3, ','  )) )as tname
from(select distinct isbn10,author, fname,lname, trim(trailing ',' from
substr( temp2, 1,instr( temp2, ','  ) )) as mname, trim(leading ','
from substr( temp2,instr( temp2, ','  )) ) as temp3
from (
```

```
select distinct isbn10,author, fname, trim(trailing ',' from substr(
temp, 1,instr( temp, ','  ) )) as lname, trim(leading ',' from substr(
temp,instr( temp, ','  )) )as temp2
from (select distinct isbn10,author, trim(trailing ',' from substr(
author, 1,instr( author, ','  ) )) as fname, trim(leading ',' from
substr( author,instr( author, ','  )) )
as temp from project1_books_load)))) tab1
on au.author_name=tab1.fname or au.author_name=tab1.lname or
au.author_name=mname or au.author_name=pname or au.author_name=tname);
```

project1_publisher:

```
insert into project1_publisher (publisher_name)  (select
distinct(publisher) from project1_books_load );
```

project1_book_publisher:

```
insert into project1_book_publisher (publisher_id,isbn10) (select
pp.publisher_id, pbl.isbn10 from project1_publisher pp full outer join
project1_books_load pbl on pp.publisher_name=pbl.publisher
where pp.publisher_id is not null or pbl.isbn10 is not null);
```

project1_library_branch:

```
insert into
project1_library_branch(branch_id,branch_name,address)(select
branch_id,branch_name,address from project1_library_branch_load);
```

project1_borrowers:

```
insert into
project1_borrowers(cardno,ssn,first_name,last_name,email,address,city
,state,phone)
(select id0000id,ssn,first_name,last_name,email,address,city
,state,phone from project1_borrowers_load);
```

project1_book_copies:

```
insert into project1_book_copies (isbn, branch_id,no_of_copies)(select
book_id, branch_id, no_of_copies from project1_book_copies_load );

insert into project1_book_copies (isbn, branch_id,no_of_copies)(select
isbn, branch_id, 1  from project1_book_copies where no_of_copies=2 );

update project1_book_copies set no_of_copies=1 where no_of_copies=2;
```

project1_book_loans:

```
insert into project1_book_loans (cardno,
book_id,due_date,date_out,date_in) (select
cardno,book_id,due_date,date_out,date_in  from (select cardno,book_id,
to_date(
              trunc(
                   dbms_random.value(to_char(date '2018-01-01','j')
                                     ,to_char(date '2019-12-31','j')
                                     )
                    ),'j'
```

```
                ) as due_date,  to_date(
               trunc(
                    dbms_random.value(to_char(date '2018-01-01','j')
                                     ,to_char(date '2019-12-31','j')
                                     )
                     ),'j'
                ) as date_out , to_date(
               trunc(
                    dbms_random.value(to_char(date '2018-01-01','j')
                                     ,to_char(date '2019-12-31','j')
                                     )
                     ),'j'
                ) as date_in from(select cardno from project1_borrowers
order by dbms_random.value fetch next 2500 rows only),
(select book_id from project1_book_copies order by dbms_random.value
fetch next 2500 rows only) order by dbms_random.value fetch next 2000
rows only) where date_out < due_date and date_out<date_in);
```

project1_book_fines:

```
insert into project1_fines (loan_id,fine_amt,paid)(select loan_id, case
when delay*1.50 <100 then delay*1.50 else 100.00 end as fine_amt,
round(dbms_random.value(0,1)) as paid
from (select distinct loan_id, date_in-due_date as delay  from
project1_book_loans ));
```

Book Search and Availability:

```
select project1_books.isbn10,project1_books.title,project1_authors.author_name,case when
project1_book_loans.date_in<=sysdate or project1_book_loans.date_in is null then 'available' else
'not avialable' end as availablity,project1_book_copies.branch_id

from project1_books left outer join project1_book_authors

on project1_books.isbn10=project1_book_authors.isbn10

left outer join project1_authors

on project1_book_authors.author_id=project1_authors.author_id

left outer join project1_book_copies

on project1_book_copies.isbn=project1_books.isbn10

left outer join project1_book_loans

on project1_book_copies.book_id=project1_book_loans.book_id

where lower(project1_books.isbn10) like lower('%%') and project1_book_copies.branch_id like '%%'
and (lower(project1_books.title) like lower('%will%') or lower(project1_authors.author_name)  like
lower('%will%')) ;
```

Note: You can add constraint at appropriate positions in the above select query.

One such example of keyword ‘will’ is shown above.

Report:

Top 10 books based on delay:

```
select isbn10,title,delay_time from(
select isbn10,title ,avg(delay) as delay_time from(
select project1_book_copies.book_id as id,project1_books.isbn10 as
isbn10,project1_books.title as title,project1_book_loans.date_in-
project1_book_loans.due_date as delay
from project1_books left outer join project1_book_authors
on project1_books.isbn10=project1_book_authors.isbn10
left outer join project1_authors
on project1_book_authors.author_id=project1_authors.author_id
left outer join project1_book_copies
on project1_book_copies.isbn=project1_books.isbn10
left outer join project1_book_loans
on project1_book_copies.book_id=project1_book_loans.book_id)
where delay is not null and delay>=0
group by isbn10,title
order by avg(delay) desc)
where rownum<11;
```

Top 10 books based on no of copies across all locations:

```
select isbn10, title , count_of_books from
(select pb.isbn10 as isbn10, pb.title as title,count(pb.isbn10) as
count_of_books
from project1_books pb
inner join project1_book_copies pbc
on pb.isbn10=pbc.isbn
inner join project1_book_loans pbl
on pbc.book_id= pbl.book_id
group by pb.isbn10,pb.title
order by count_of_books desc) where rownum<11;
```