Aarushi Agarwal
UTD ID: 2021460509

Project 1 Report

ER diagram

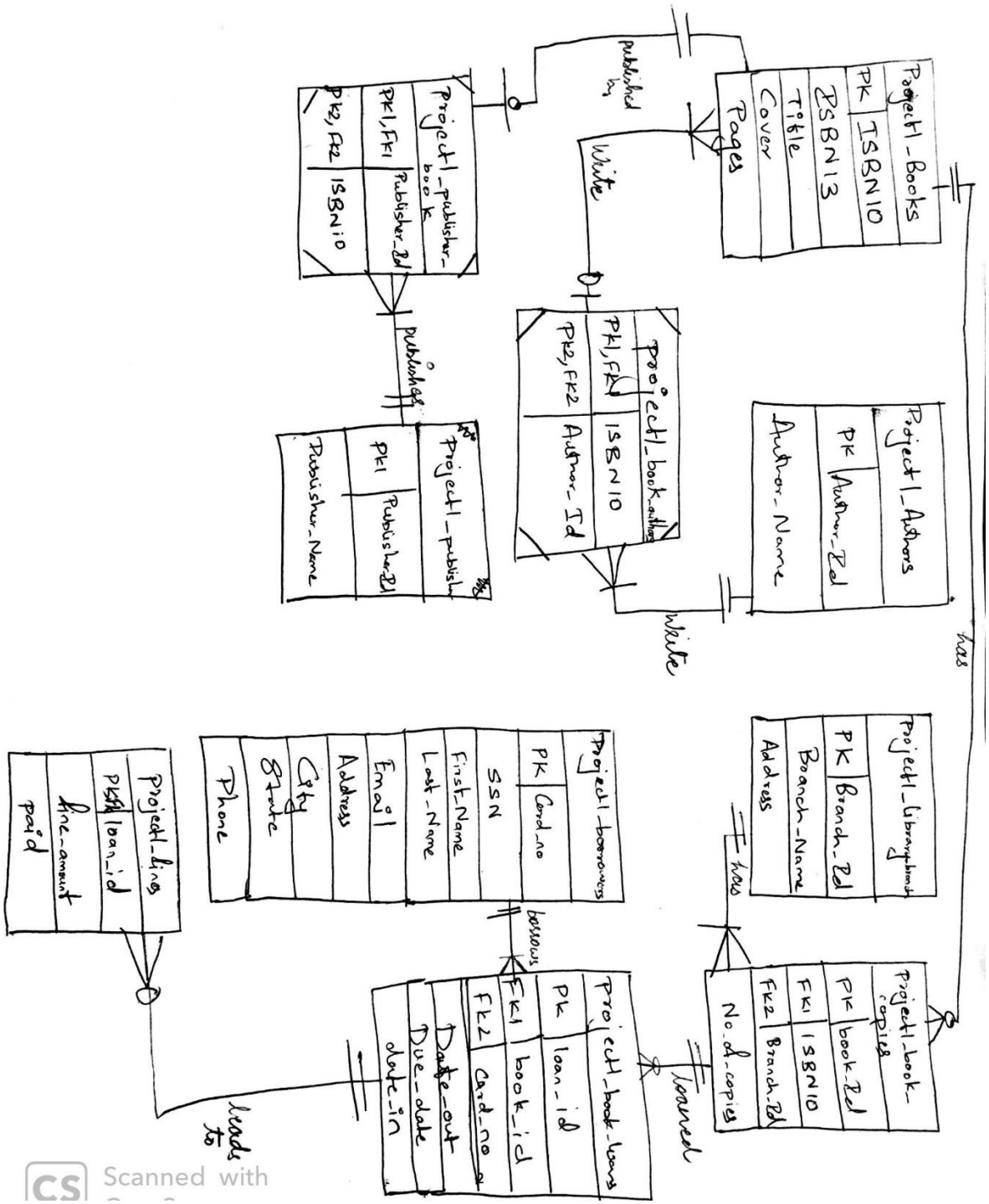

Figure 1:ER Diagram

Create table statements:

Queries 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(250), Cover varchar2(300), Publisher varchar2(150), No_Of_Pages number(6));

Create Statement for Project1_books_copies_load

create table project1_book_copies_load(book_id varchar(10), branch_id number(1), no_of_copies number(1));

Create Statement for Project1_borrowers_load

create table project1_borrowers_load (ID0000id VARCHAR2(8),ssn VARCHAR2(12),first_name VARCHAR2(100),last_name VARCHAR2(100),email VARCHAR2(200),address VARCHAR2(200),city VARCHAR2(150),state VARCHAR2(5),phone VARCHAR2(14));

Create Statement for Project1_library_branch_load

create table project1_library_branch_load(branch_id number(2),branch_name varchar2(50),address varchar2(100));

Queries for normalized tables

Create Statement for Books Table

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

Create Statement for Authors Table

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

Create Statement for Books and Authors mapping table Table

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

Create Statement for Library Branch Table

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

Create Statement for Book Copies Table

create table project1_book_copies(book_id integer GENERATED BY DEFAULT AS IDENTITY (START WITH 1) NOT NULL PRIMARY KEY,isbn10 varchar2(10), branch_id number(2), no_of_copies number(2));

Create Statement for Borrowers Table

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 Statement for Book Loans Table

```
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 Statement for Fines Table

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

Create Statement for Publisher Table

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

Create Statement for Book Publisher Table

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

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_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(isbn10) references project1_books(isbn10);
```

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

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);

SQL statements for loading data in normalized tables

Insert statement for Project1_books table

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

Insert statement for Project1_authors table

Authors table is loaded from project1_BOOKS_LOAD. In this table, Author is a multi-valued attribute, so to make it atomic or bring the table in 1NF form we used the below query to separate author names into different columns then loaded the data into a temporary table (tempauthor). Ran this query 5 times considering one column at a time from (fname,lname,mname, pname, tname) to load entries in temauthor's author_name column.

select DISTINCT author, fname,lname,mname, Trim(trailing ',' from SUBSTR(endgame3, 1,INSTR(endgame3, ','))) AS PNAME, Trim(leading ',' from SUBSTR(endgame3,INSTR(endgame3, ',')))AS TNAME from (select DISTINCT ISBN10,author, fname,lname, Trim(trailing ',' from SUBSTR(endgame2, 1,INSTR(endgame2, ','))) as mname, Trim(leading ',' from SUBSTR(endgame2,INSTR(endgame2, ','))) as endgame3 from (select DISTINCT ISBN10,author, fname, Trim(trailing ',' from SUBSTR(endgame, 1,INSTR(endgame, ','))) as lname, Trim(leading ',' from SUBSTR(endgame,INSTR(endgame, ',')))as endgame2 from (select DISTINCT ISBN10,author, Trim(trailing ',' from SUBSTR(author, 1,INSTR(author, ','))) as fname, Trim(leading ',' from SUBSTR(author,INSTR(author, ','))) as endgame from project1_BOOKS_LOAD)));

Used the temporary table created to load data into project1_authors table

INSERT INTO project1_authors(author_name)(select distinct author_name from tempauthor where author_name is not null);

Insert statement for Project1_book_authors table

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(endgame3, 1,INSTR(endgame3, ','))) AS PNAME, Trim(leading ',' from SUBSTR(endgame3,INSTR(endgame3, ',')))AS TNAME

from(select DISTINCT ISBN10,author, fname,lname, Trim(trailing ',' from SUBSTR(endgame2, 1,INSTR(endgame2, ','))) as mname, Trim(leading ',' from SUBSTR(endgame2,INSTR(endgame2, ','))) as endgame3 from (select DISTINCT ISBN10,author, fname, Trim(trailing ',' from SUBSTR(endgame,

1,INSTR(endgame, ','))) as lname, Trim(leading ',' from SUBSTR(endgame,INSTR(endgame, ',')))as endgame2 from (select DISTINCT ISBN10,author, Trim(trailing ',' from SUBSTR(author, 1,INSTR(author, ','))) as fname, Trim(leading ',' from SUBSTR(author,INSTR(author, ','))) as endgame 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);

Insert statement for Project1_ publisher table

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

Insert statement for Project1_ book_publisher table

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);

Insert statement for Project1_ library_branch table

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

Insert statement for Project1_ Borrowers table

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);

Insert statement for Project1_Book_Copies table

INSERT INTO PROJECT1_BOOK_COPIES (ISBN10, BRANCH_ID,NO_OF_COPIES)(SELECT Book_Id, BRANCH_ID, No_Of_Copies FROM PROJECT1_BOOK_COPIES_Load);

Implemented individualization by creating another entry for books with 2 copies. Hardcoded value to 2 here because it was the maximum number of copies.

INSERT INTO PROJECT1_BOOK_COPIES (ISBN10, BRANCH_ID,NO_OF_COPIES)(SELECT ISBN10, BRANCH_ID, 1 As No_Of_Copies FROM PROJECT1_BOOK_COPIES WHERE No_Of_Copies=2);

UPDATE PROJECT1_BOOK_COPIES SET NO_OF_COPIES=1 WHERE No_Of_Copies=2;

Insert statement for Project1_Book_Loans table

SQL statement to generate At least 500 books check-outs for at least 200 different borrowers and 100 different books:

```
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 2000 rows only),(select book_id from project1_book_copies order by dbms_random.value fetch next 2000 rows only) order by dbms_random.value fetch next 2000 rows only) where date_out < due_date and date_out<date_in);
```

Insert statement for Project1_Fines table

SQL statement to generate at least 50 fines records for 20 different borrowers

```
INSERT INTO project1_fines (loan_Id,Fine_Amt,Paid)(SELECT LOAN_ID, CASE WHEN DELAY*1.20 <100 THEN DELAY*1.20 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

Query for Book Search and Availability functionality. User would have to specify the values for branch_id or isbn10 or names for book or author. Fields have been left blank with wild card checking so those will be populated as per user search query. An example would be ‘will’ which has been populated in the below query.

```
select pb.isbn10,pb.title ,pa.author_name, case when date_in<= sysdate then 'available' when date_in is null then 'available' else 'unavailable' end as availability
```

```
from project1_books pb
left join
project1_book_authors pba on pb.isbn10 = pba.isbn10
left join project1_authors pa
on pa.author_id=pba.author_id
left join project1_book_copies pbc
on pb.isbn10=pbc.isbn10
left join project1_book_loans pbl
on pbl.book_id=pbc.book_id where pbc.branch_id like '%%'
and lower(pb.isbn10) like lower('%%') and (lower(pb.title) like lower('%will%') or
lower(pa.author_name) like lower('%will%')) ;
```

Reports

Top 10 books based on the number of days it was rent out

```
select isbn10,title,delay_time from(
select isbn10,title ,avg(delay) as delay_time from(
select pbc.book_id as id,pb.isbn10 as isbn10,pb.title as title,pbl.date_in-pbl.due_date as delay
from project1_books pb left outer join project1_book_authors pba
on pb.isbn10=pba.isbn10
left outer join project1_authors pa
on pba.author_id=pa.author_id
left outer join project1_book_copies pbc
on pbc.isbn10=pb.isbn10
left outer join project1_book_loans pbl
on pbc.book_id=pbl.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 the number of copies of books present 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.isbn10

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;
```