Name: Aashish Reddy Vundhyala

UTD ID: 2021566387

All the tables are created in Project1 Database in my account.

1.) DESIGN AND DB ARCHITECTURE

1.) b.) Create statements

drop table project1\_books;

drop table project1\_authors;

drop table project1\_book\_authors;

drop table project1\_library\_branch;

drop table project1\_fines;

drop table project1\_book\_loans;

drop table project1\_borrowers;

drop table project1\_book\_copies;`

create table project1\_books

(ISBN10 VARCHAR2(26) NOT NULL,

ISBN13 VARCHAR2(38),

Title VARCHAR2(256),

Author VARCHAR2(128),

Cover VARCHAR2(128),

Publisher VARCHAR2(128),

pages INTEGER,

CONSTRAINT isbn10\_pk PRIMARY KEY(ISBN10) );

create table project1\_authors

(Author\_id NUMBER NOT NULL,

Name varchar2(500),

CONSTRAINT authorid\_pk PRIMARY KEY(Author\_id));



create table project1\_book\_authors

(Author\_id NUMBER NOT NULL,

ISBN10 VARCHAR2(26) NOT NULL,

Author varchar2(128),

CONSTRAINT authorisbn\_pk PRIMARY KEY(Author\_id, ISBN10),

CONSTRAINT authorid\_fk FOREIGN KEY(Author\_id) REFERENCES project1\_authors(Author\_id),

CONSTRAINT isbn10\_fk FOREIGN KEY(ISBN10) REFERENCES project1\_books(ISBN10) );



create table project1\_library\_branch

(branch\_id INTEGER NOT NULL,

branch\_name VARCHAR2(100),

address VARCHAR2(500),

zip VARCHAR2(10),

CONSTRAINT branchid\_pk PRIMARY KEY(branch\_id));



create table project1\_book\_copies

(book\_id VARCHAR2(26) NOT NULL,

branch\_id INTEGER NOT NULL,

no\_of\_copies INTEGER,

ISBN10 VARCHAR2(26),

CONSTRAINT bookid\_branchid\_pk PRIMARY KEY(book\_id),

CONSTRAINT branchid\_fkk FOREIGN KEY(branch\_id) REFERENCES project1\_library\_branch(branch\_id),

CONSTRAINT isbn10\_fkkk FOREIGN KEY(ISBN10) REFERENCES project1\_books(ISBN10) );



create table project1\_borrowers(

card\_no VARCHAR2(26) NOT NULL,

ssn VARCHAR2(26),

first\_name VARCHAR2(26),

last\_name VARCHAR2(26),

email VARCHAR2(128),

address VARCHAR2(128),

city VARCHAR2(26),

state VARCHAR2(26),

phone varchar2(26),

CONSTRAINT cardno\_pk PRIMARY KEY(card\_no));



create table project1\_book\_loans(

loan\_id NUMBER NOT NULL,

book\_id VARCHAR2(26) NOT NULL,

card\_no VARCHAR2(26) NOT NULL,

Date\_out DATE,

Date\_in DATE,

Due\_date DATE,

branch\_id INTEGER NOT NULL,

CONSTRAINT loanid\_pk PRIMARY KEY(loan\_id),

CONSTRAINT bookid\_fk FOREIGN KEY(book\_id) REFERENCES project1\_book\_copies(book\_id),

CONSTRAINT cardno\_fk FOREIGN KEY(card\_no) REFERENCES project1\_borrowers(card\_no) );

ALTER TABLE project1\_book\_loans

ADD CONSTRAINT branch\_id\_fkk FOREIGN KEY(branch\_id) REFERENCES project1\_library\_branch(branch\_id);



create table project1\_fines(

loan\_id NUMBER NOT NULL,

fine\_amt NUMBER,

paid VARCHAR2(20),

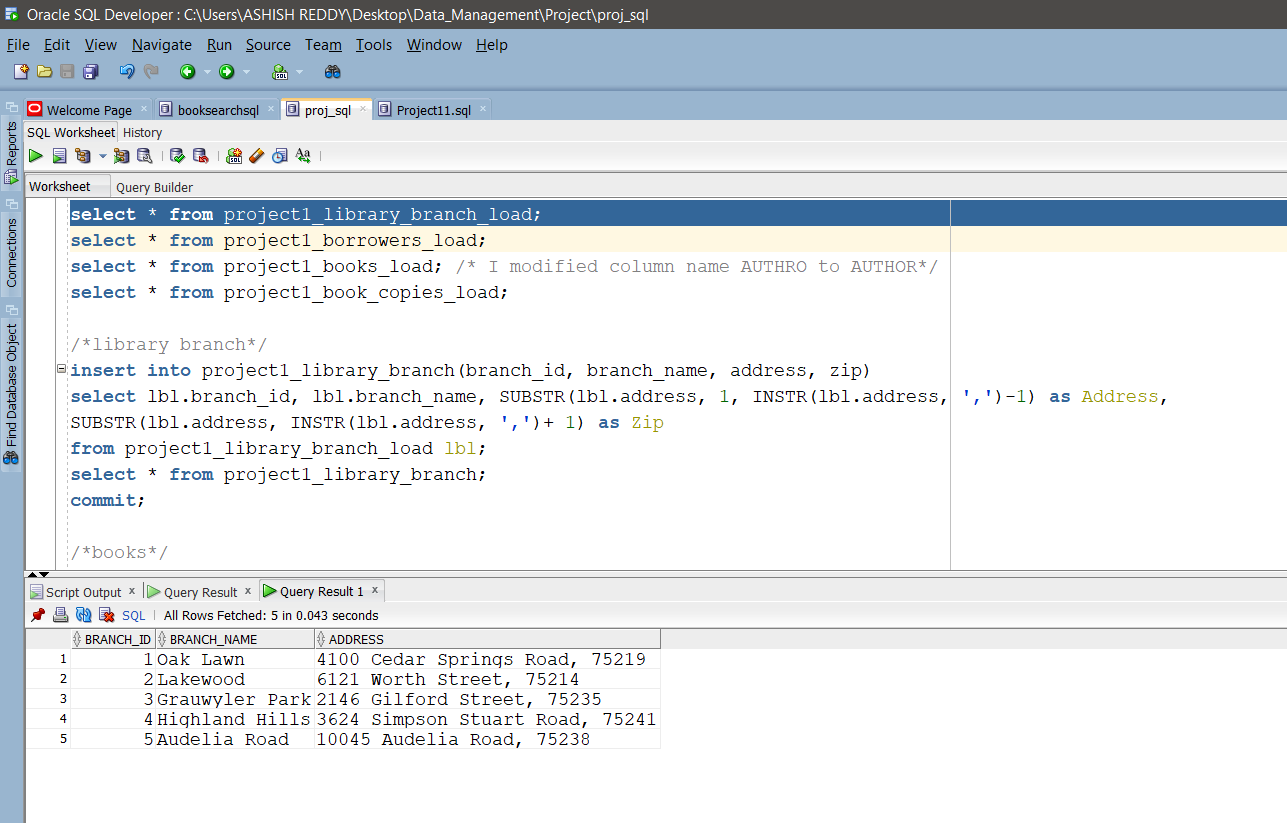
CONSTRAINT loanid\_fk FOREIGN KEY(loan\_id) REFERENCES project1\_book\_loans(loan\_id));



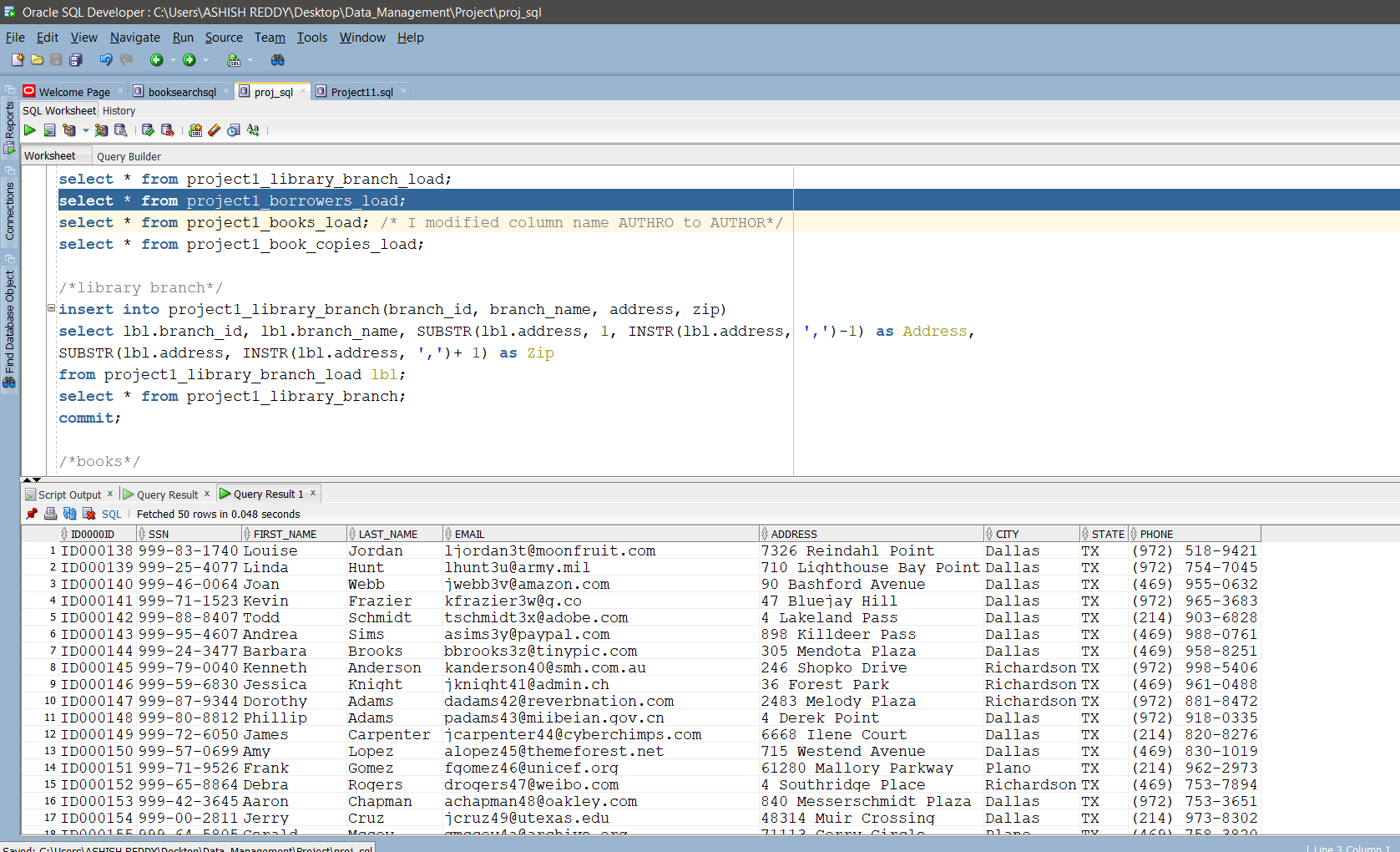
2.) Data Load, Normalization and data generation

2.) 1.) Initial Load files to tables.

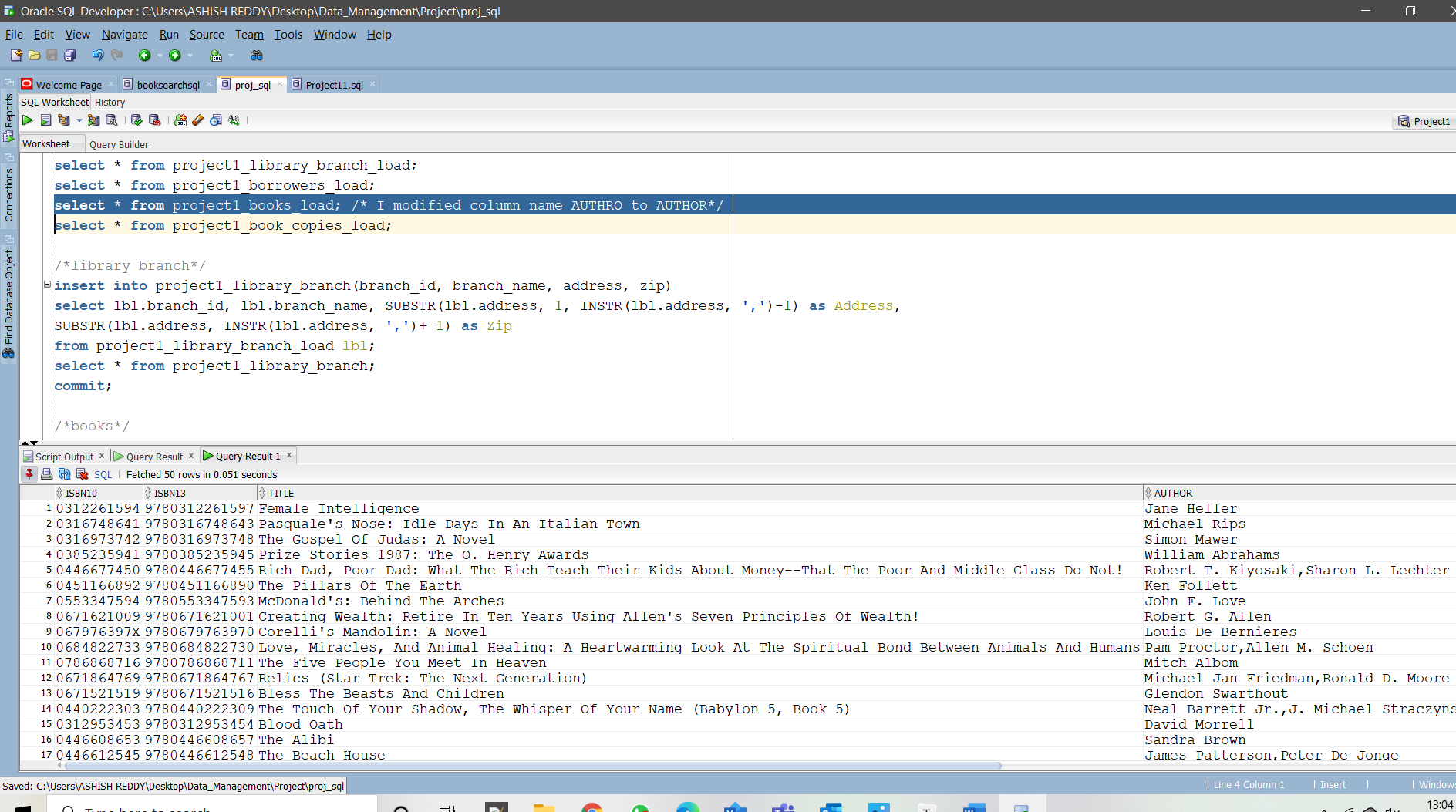
select \* from project1\_library\_branch\_load;



select \* from project1\_borrowers\_load;

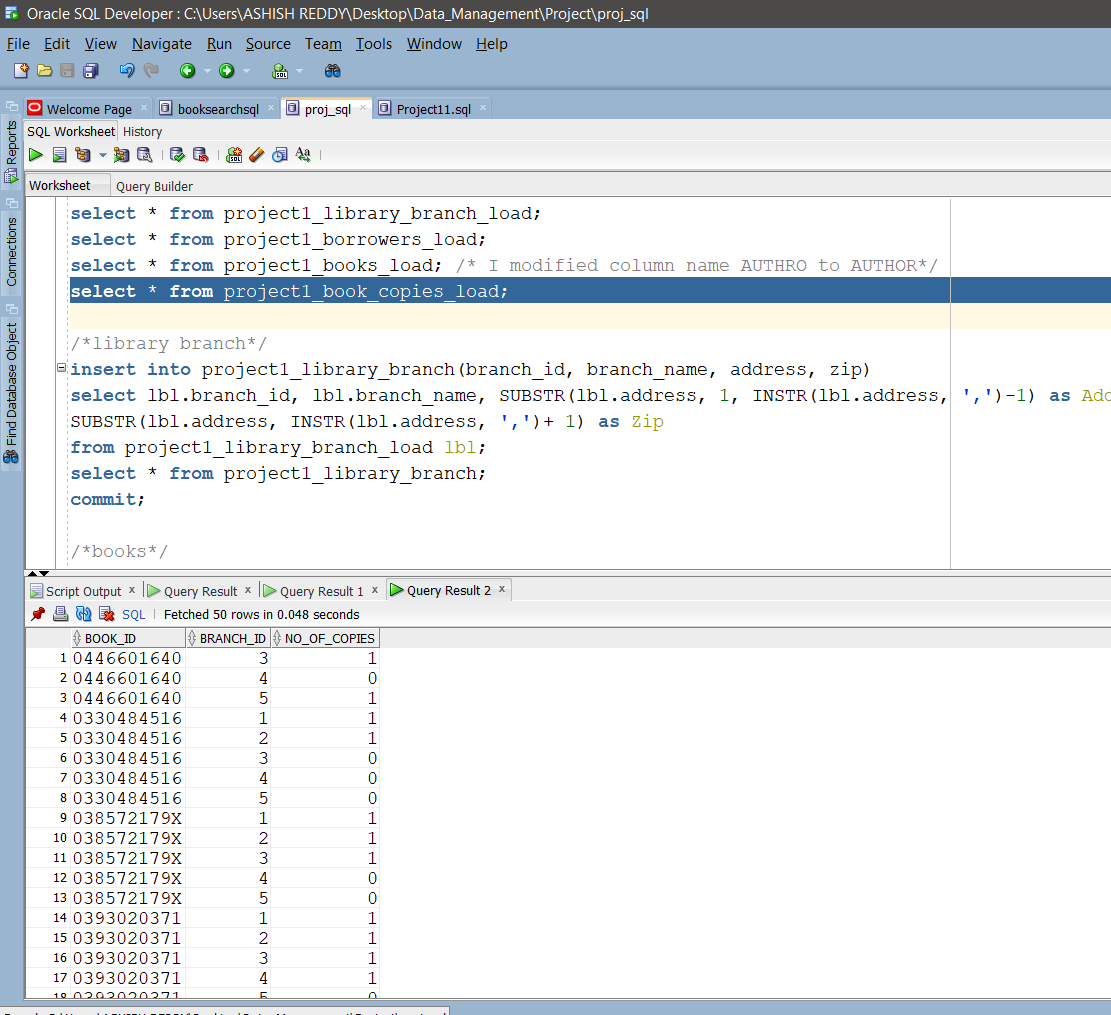


select \* from project1\_books\_load;



/\*I modified column name AUTHRO to AUTHOR \*/

select \* from project1\_book\_copies\_load;



2.) 2.) SQL STATEMENTS ( Normalized tables)

/\* library branch \*/

insert into project1\_library\_branch(branch\_id, branch\_name, address, zip)

select lbl.branch\_id, lbl.branch\_name,

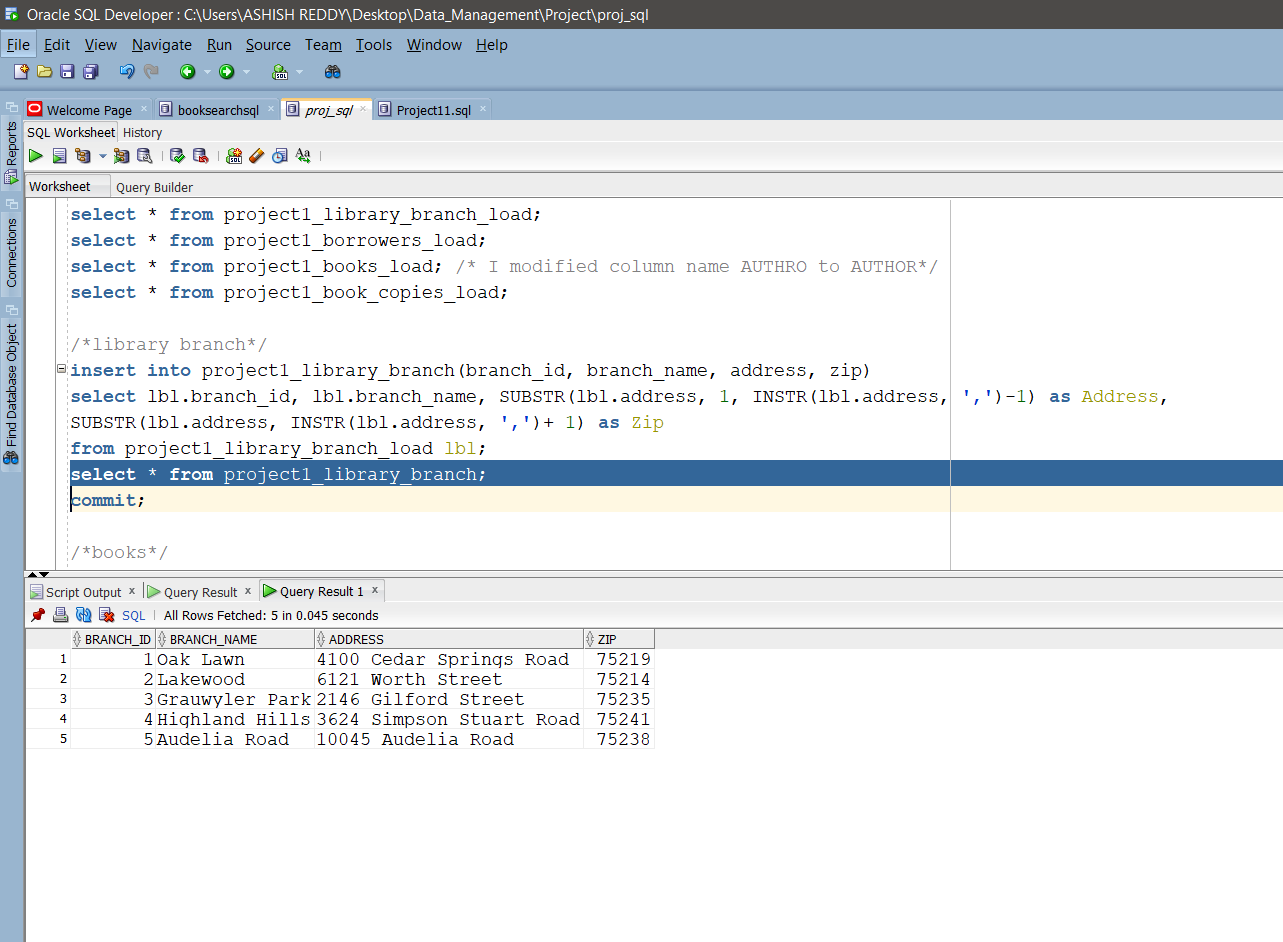
SUBSTR(lbl.address, 1, INSTR(lbl.address, ',') -1) as Address,

SUBSTR(lbl.address, INSTR(lbl.address, ',')+ 1) as Zip from

project1\_library\_branch\_load lbl;

select \* from project1\_library\_branch;

commit;



/\* books \*/

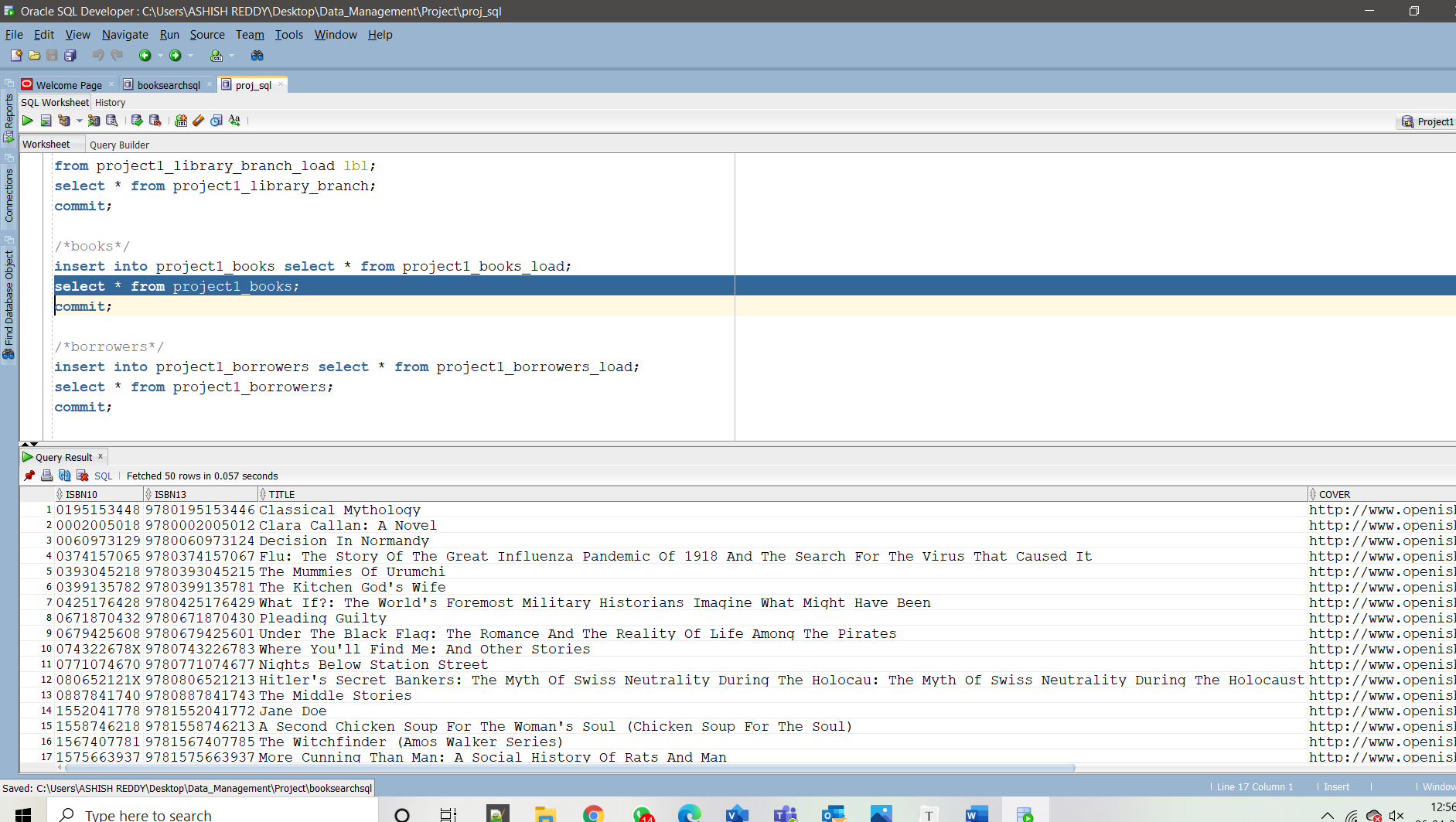
insert into project1\_books

select \* from project1\_books\_load;

/\* dropping Author column at the end\*/

select \* from project1\_books;

commit;



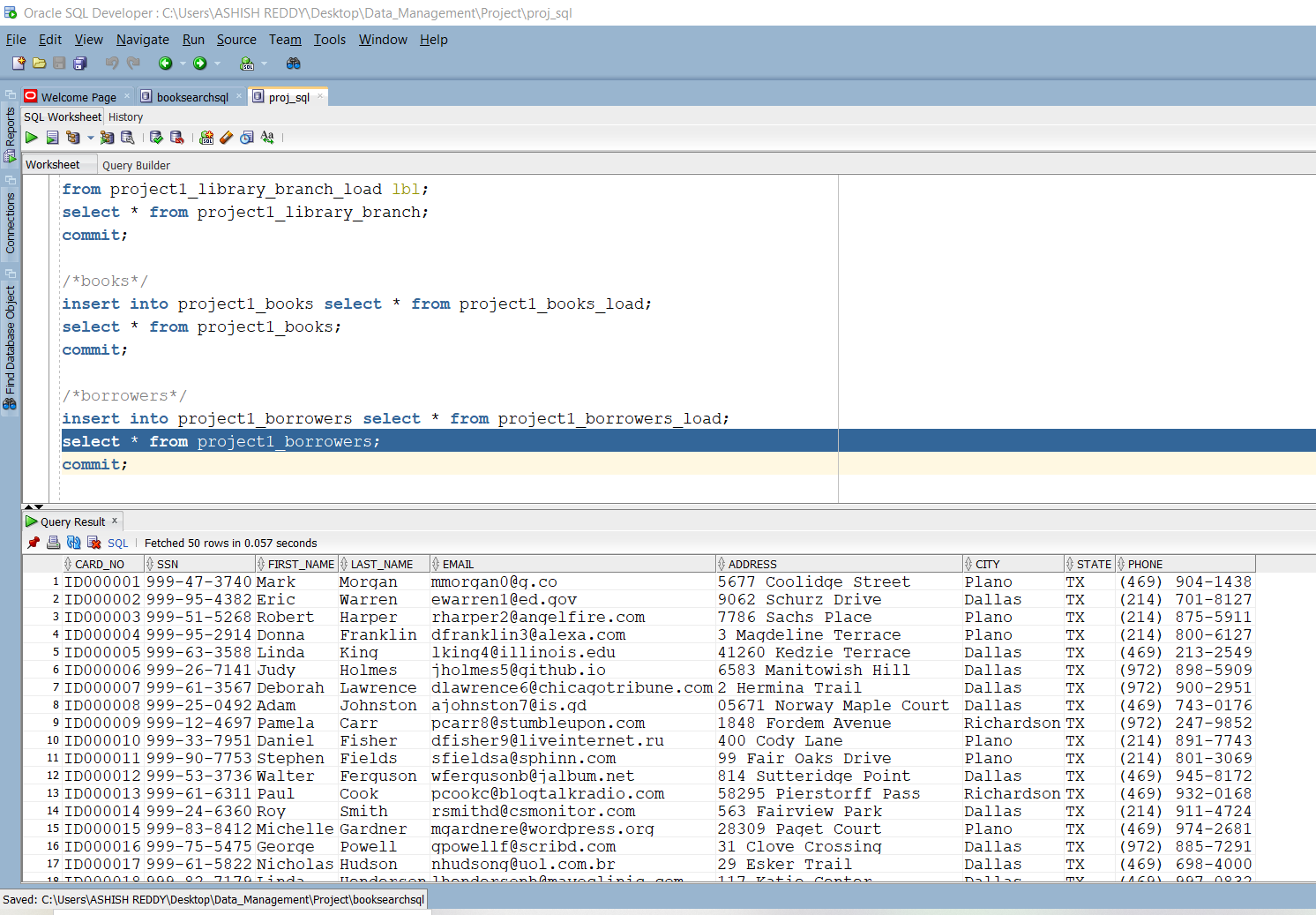
/\*borrowers \*/

insert into project1\_borrowers

select \* from project1\_borrowers\_load;

select \* from project1\_borrowers;

commit;



/\*book\_copies \*/

insert into project1\_book\_copies(book\_id, branch\_id, no\_of\_copies, ISBN10)

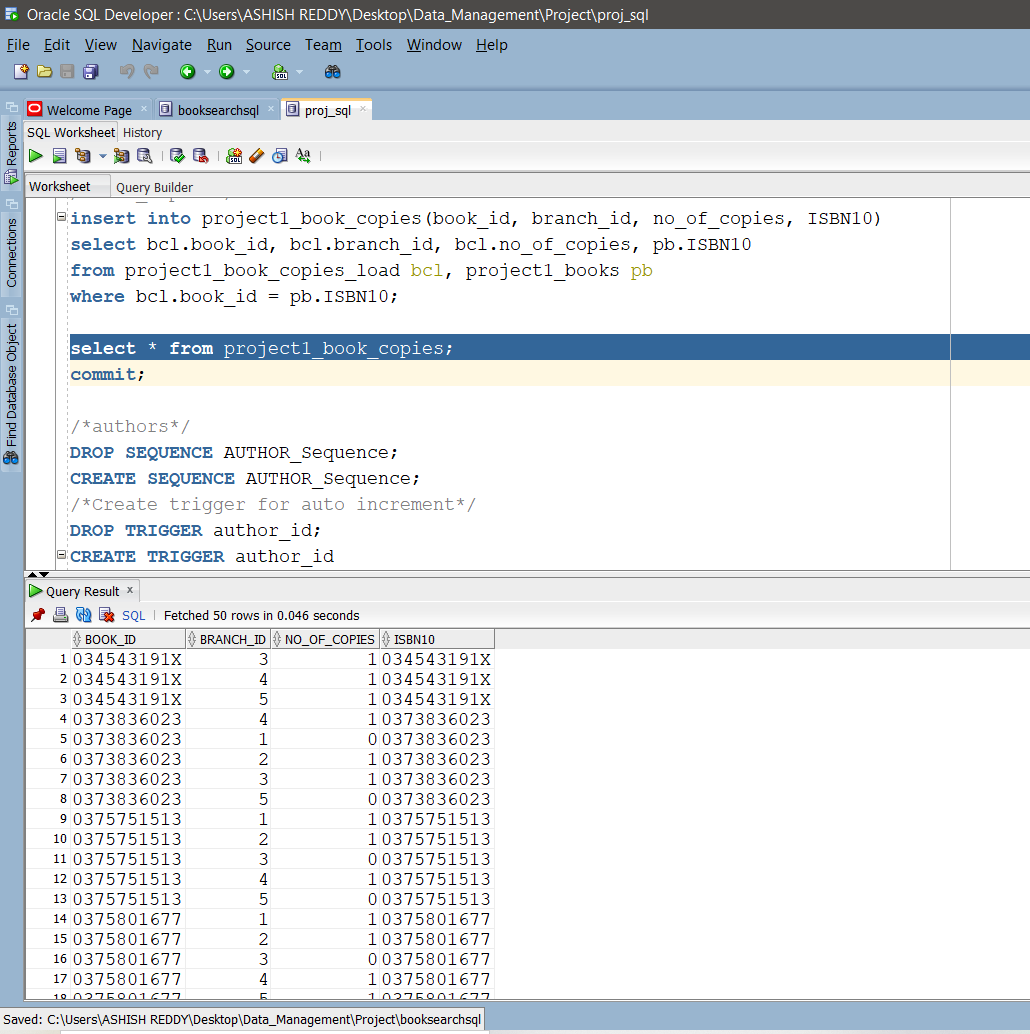
select bcl.book\_id, bcl.branch\_id, bcl.no\_of\_copies, pb.ISBN10

from project1\_book\_copies\_load bcl, project1\_books pb

where bcl.book\_id = pb.ISBN10;

select \* from project1\_book\_copies;

commit;



/\*authors \*/

DROP SEQUENCE AUTHOR\_Sequence;

CREATE SEQUENCE AUTHOR\_Sequence;

/\* AUTO INCREMENT \*/

DROP TRIGGER author\_id;

CREATE TRIGGER author\_id

BEFORE INSERT ON PROJECT1\_AUTHORS

FOR EACH ROW

BEGIN

SELECT AUTHOR\_Sequence.nextval

INTO :new.author\_id FROM dual;

END;

insert into project1\_authors(Name)

select distinct trim(regexp\_substr(author,',+', 1, level)) author

from project1\_books\_load

where author is not null

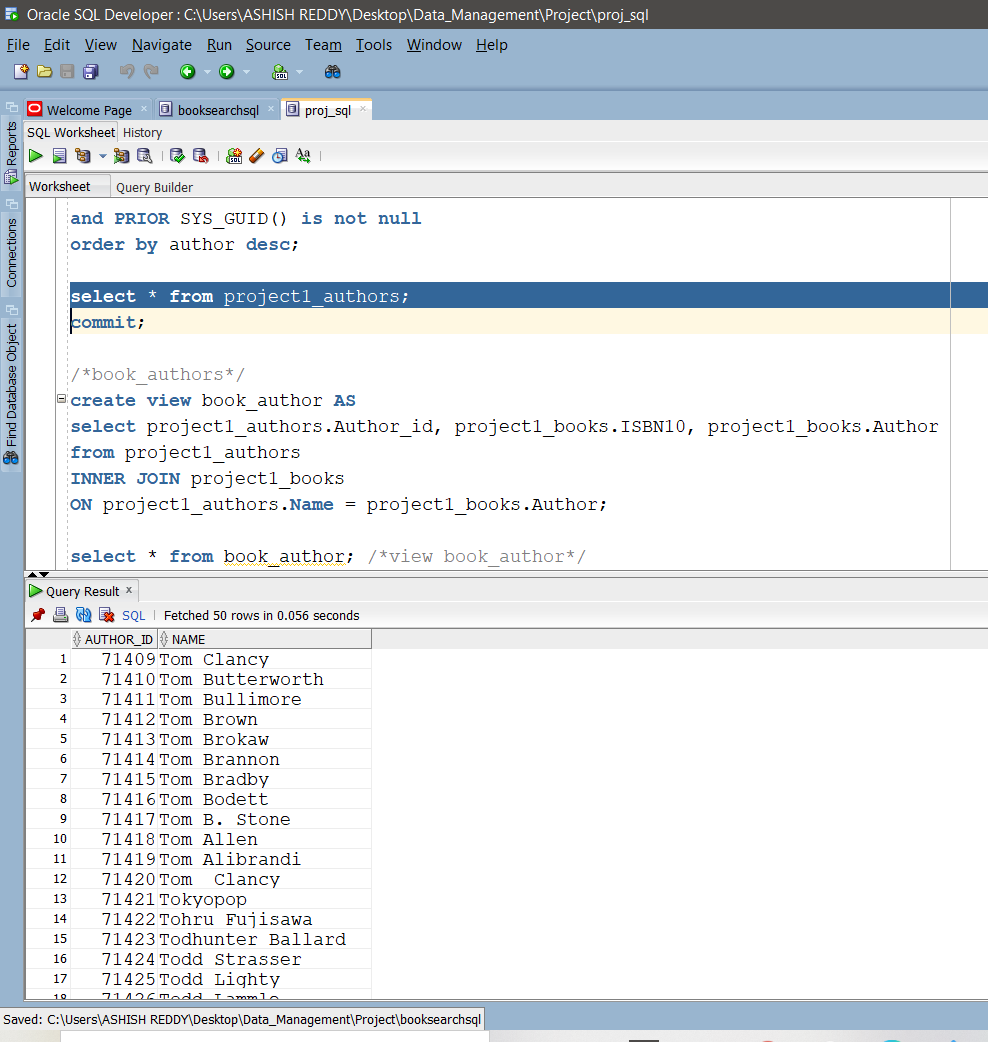
connect by regexp\_substr(author, ',+', 1, level) is not null

and PRIOR ISBN10 = ISBN10 and PRIOR SYS\_GUID() is not null

order by author desc;

select \* from project1\_authors;

commit;



/\*book\_authors \*/

create view book\_author AS

select project1\_authors.Author\_id, project1\_books.ISBN10, project1\_books.Author from project1\_authors INNER JOIN project1\_books ON

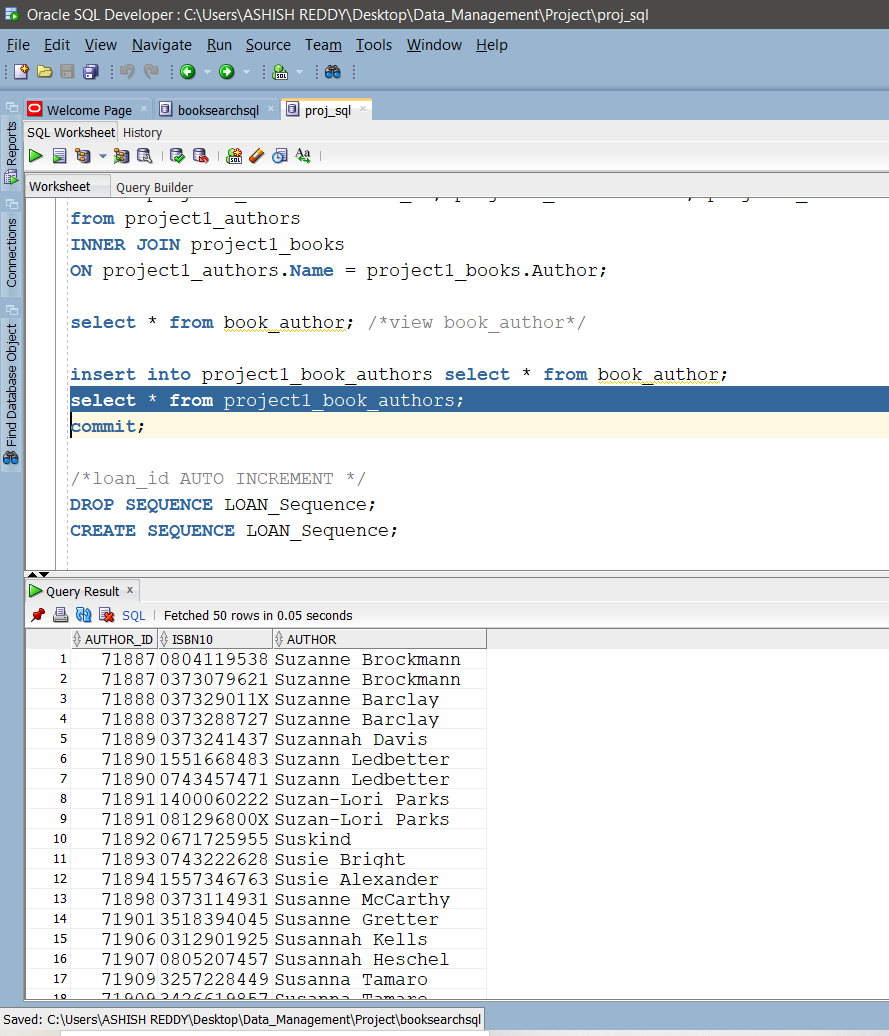
project1\_authors.Name = project1\_books.Author;

select \* from book\_author; /\*VIEW \*/

insert into project1\_book\_authors select \* from book\_author;

select \* from project1\_book\_authors;

commit;



/\* Dropping Author column in project1\_books \*/

ALTER TABLE project1\_books drop column Author;

2.) 3.)

1.) At least 500 books checkouts for at least 200 different borrowers and 100 different books.

DROP SEQUENCE LOAN\_Sequence;

CREATE SEQUENCE LOAN\_Sequence;

DROP TRIGGER loan\_id; /\* loan\_id Auto increment \*/

CREATE TRIGGER loan\_id

BEFORE INSERT ON PROJECT1\_BOOK\_LOANS

FOR EACH ROW BEGIN

SELECT LOAN\_Sequence.nextval

INTO :new.loan\_id FROM dual;

END;

insert into project1\_book\_loans(book\_id, card\_no, Date\_out, Due\_date, Date\_in, branch\_id)

select b.book\_id, a.card\_no, sysdate-20 as Date\_out, sysdate-10 as Due\_date,

sysdate - dbms\_random.value(3,12) as Date\_in, b.branch\_id

from(select \* from

(select a.card\_no, a.ssn, a.first\_name, a.last\_name, a.email, a.phone

​ from project1\_borrowers a

​ ORDER BY DBMS\_RANDOM.RANDOM)where rownum<201) a,

(select \* from (select b.book\_id, b.ISBN10, b.branch\_id

​ from project1\_book\_copies b

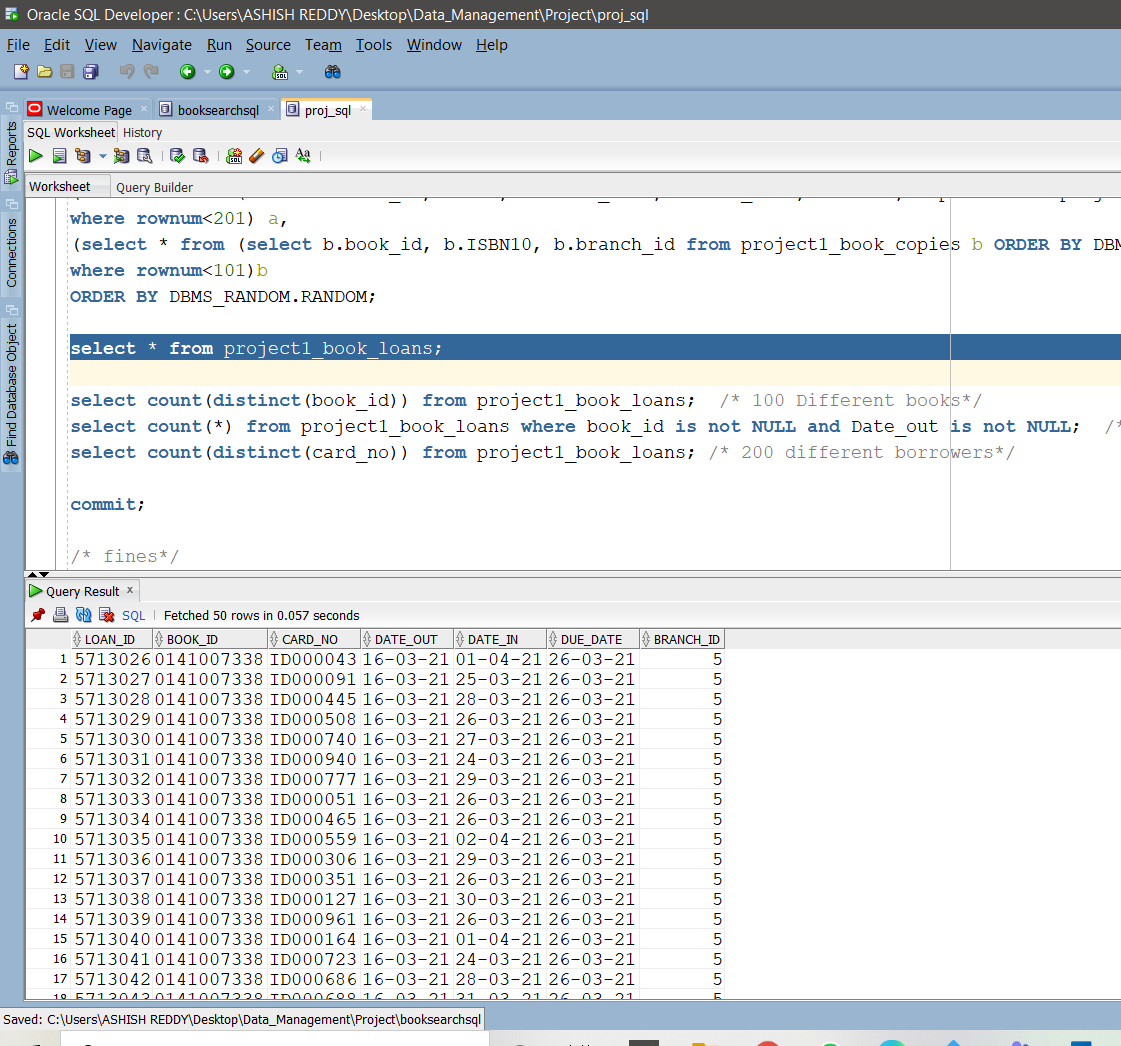
​ORDER BY DBMS\_RANDOM.RANDOM)

where rownum<101)b

ORDER BY DBMS\_RANDOM.RANDOM;

select \* from project1\_book\_loans;

commit;



/\* 100 Different books\*/

select count(distinct(book\_id)) from project1\_book\_loans;

/ \* 20000 book checkouts\*/

select count(\*) from project1\_book\_loans where book\_id is not NULL and Date\_out is not NULL;

/\* 200 different borrowers\*/

select count(distinct(card\_no)) from project1\_book\_loans;

2.) 3.)

2.) At least 50 fine records for 20 different borrowers.

insert into project1\_fines(loan\_id, fine\_amt)

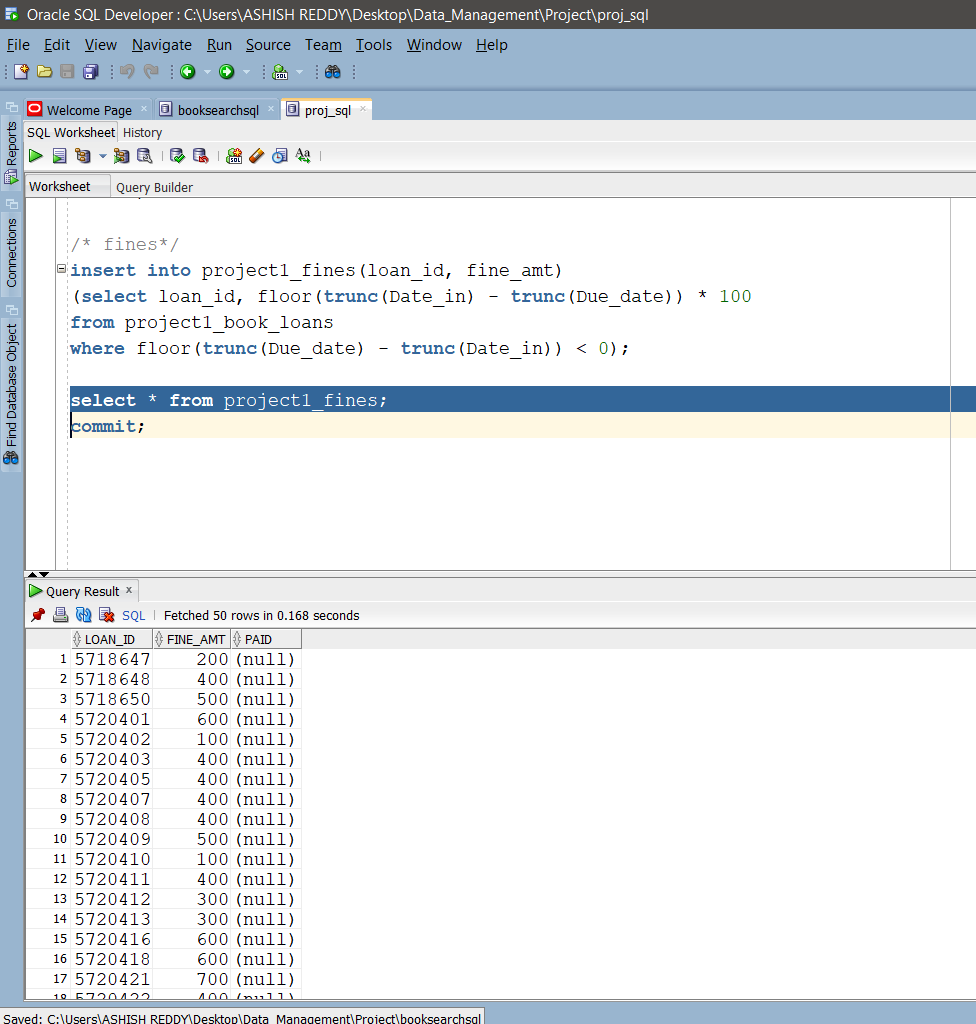
( select loan\_id, floor(trunc(Date\_in) - trunc(Due\_date)) \* 100

from project1\_book\_loans

where floor(trunc(Due\_date) - trunc(Date\_in)) < 0);

select \* from project1\_fines;

commit;



3.) Book Search and Availability

define string = 'lee';

select DISTINCT(b.ISBN10), b.title, (LISTAGG( a.NAME, ';' ) within group

(order by b.ISBN10) OVER (PARTITION BY b.ISBN10)) AS Author

from project1\_books b, project1\_Authors a, project1\_book\_authors bc

where bc.ISBN10 = b.ISBN10 and

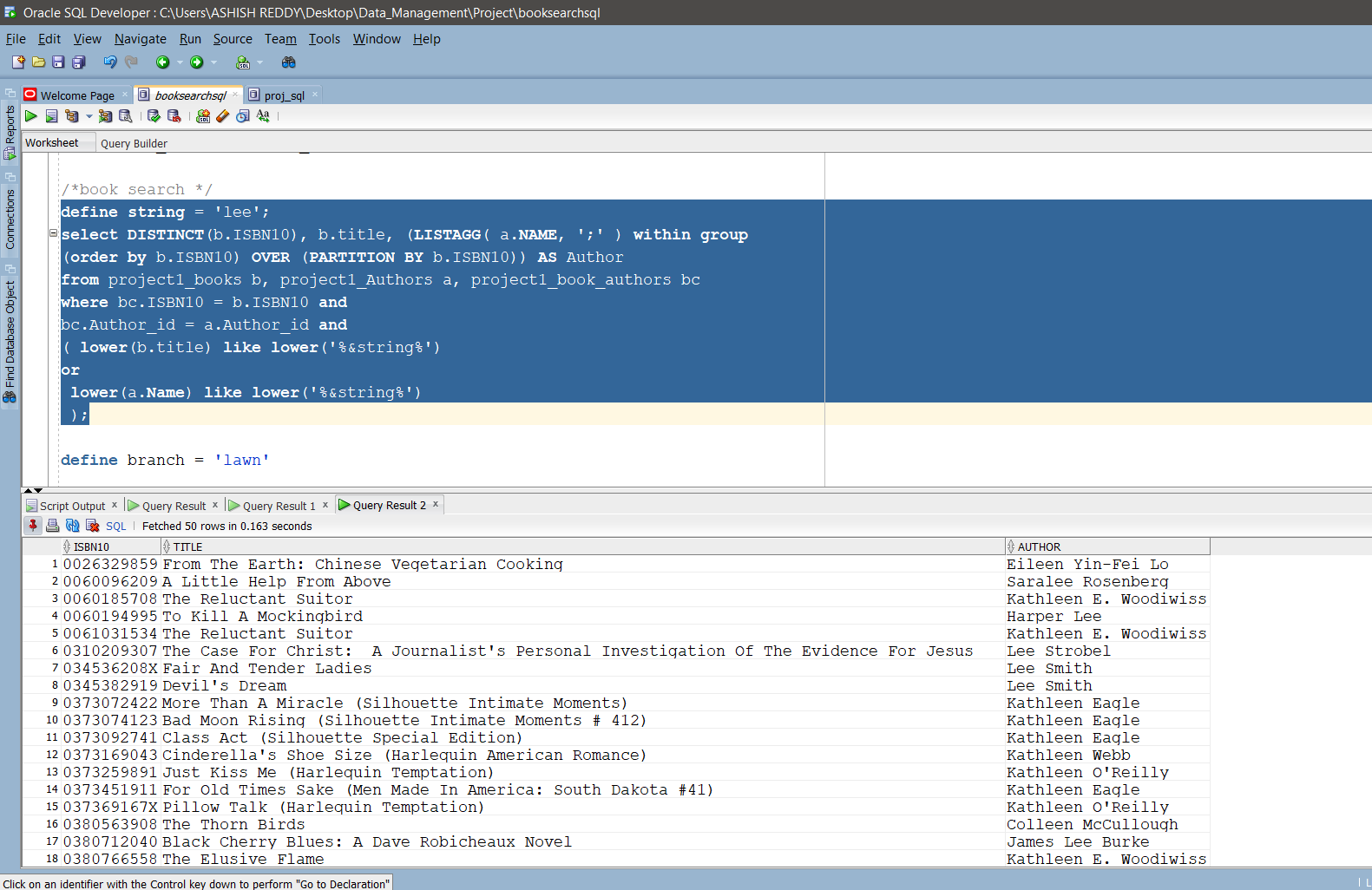
bc.Author\_id = a.Author\_id and

( lower(b.title) like lower('%&string%')

or

lower(a.Name) like lower('%&string%')

);



select distinct b.ISBN10, b.title, ba.AUTHOR, bc.no\_of\_copies, lb.branch\_name, lb.address

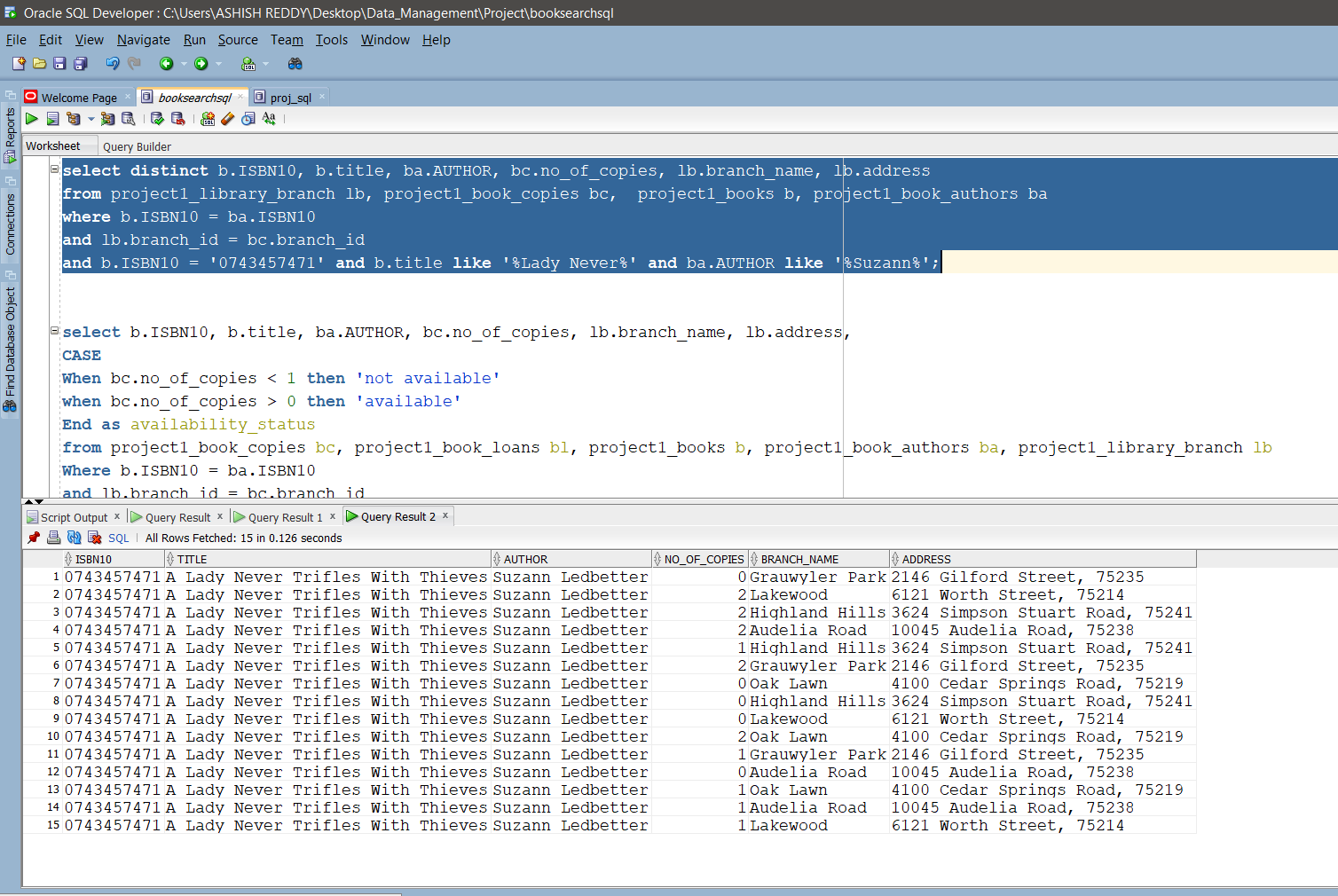
from project1\_library\_branch lb, project1\_book\_copies bc, project1\_books b, project1\_book\_authors ba

where b.ISBN10 = ba.ISBN10

and lb.branch\_id = bc.branch\_id

and b.ISBN10 = '0743457471' and

b.title like '%Lady Never%' and ba.AUTHOR like '%Suzann%';



select b.ISBN10, b.title, ba.AUTHOR, bc.no\_of\_copies, lb.branch\_name, lb.address,

CASE

When bc.no\_of\_copies < 1 then 'not available'

when bc.no\_of\_copies > 0 then 'available'

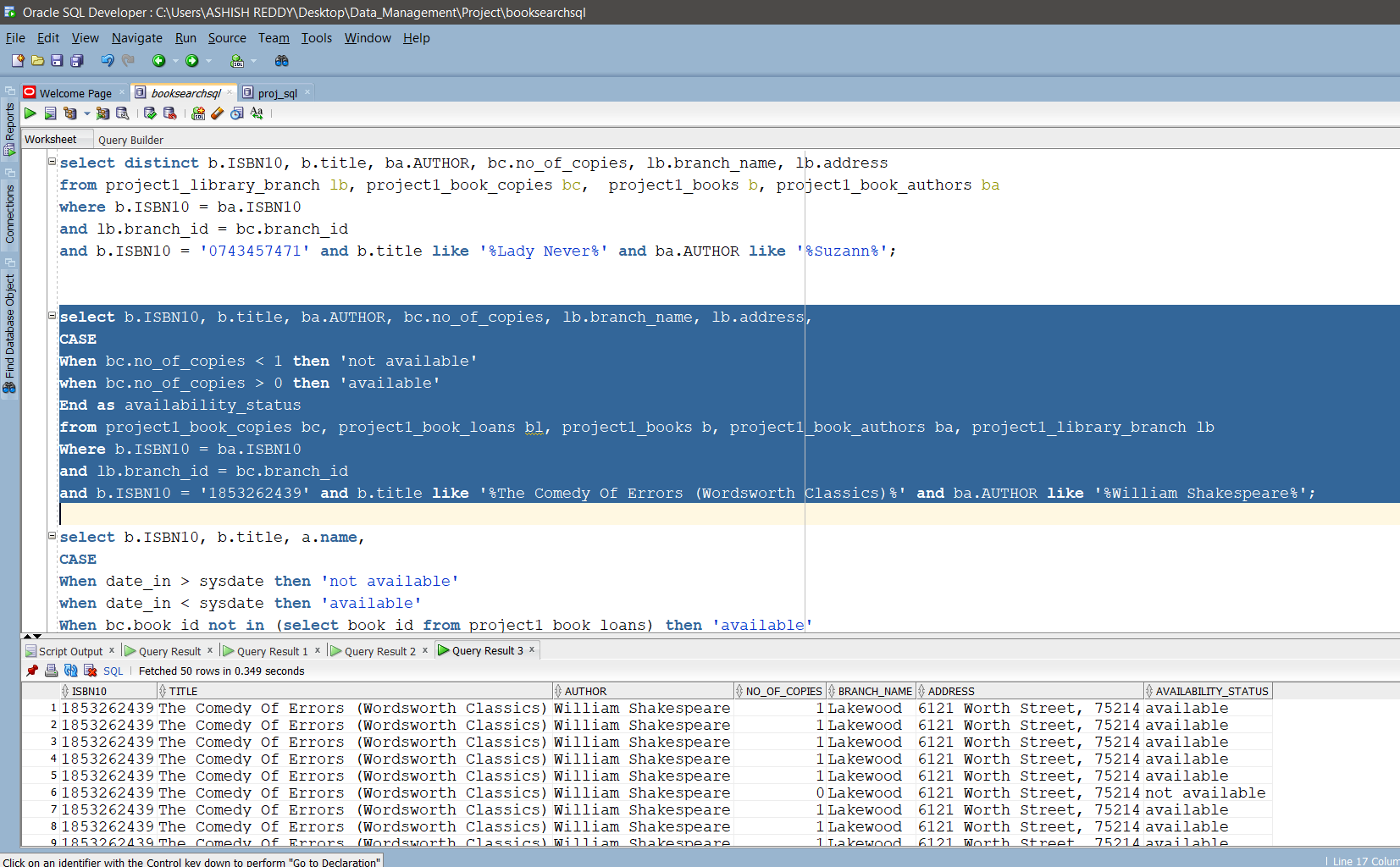
End as availability\_status

from project1\_book\_copies bc, project1\_book\_loans bl, project1\_books b, project1\_book\_authors ba, project1\_library\_branch lb

Where b.ISBN10 = ba.ISBN10

and lb.branch\_id = bc.branch\_id

and b.ISBN10 = '1853262439' and b.title like '%The Comedy Of Errors (Wordsworth Classics)%' and ba.AUTHOR like '%William Shakespeare%';



4.) Reports

/\* Report 1: Author\_id, Author Names whose book copies are exactly 6 \*/

select distinct ba.author\_id, ba.author as author

from project1\_book\_authors ba join project1\_book\_copies bc

on ba.isbn10= bc.isbn10

where bc.book\_id in

(select bookId from

(select bc.book\_id as bookId, count(bc.branch\_id) as

count\_branch, sum(no\_of\_copies) as copies

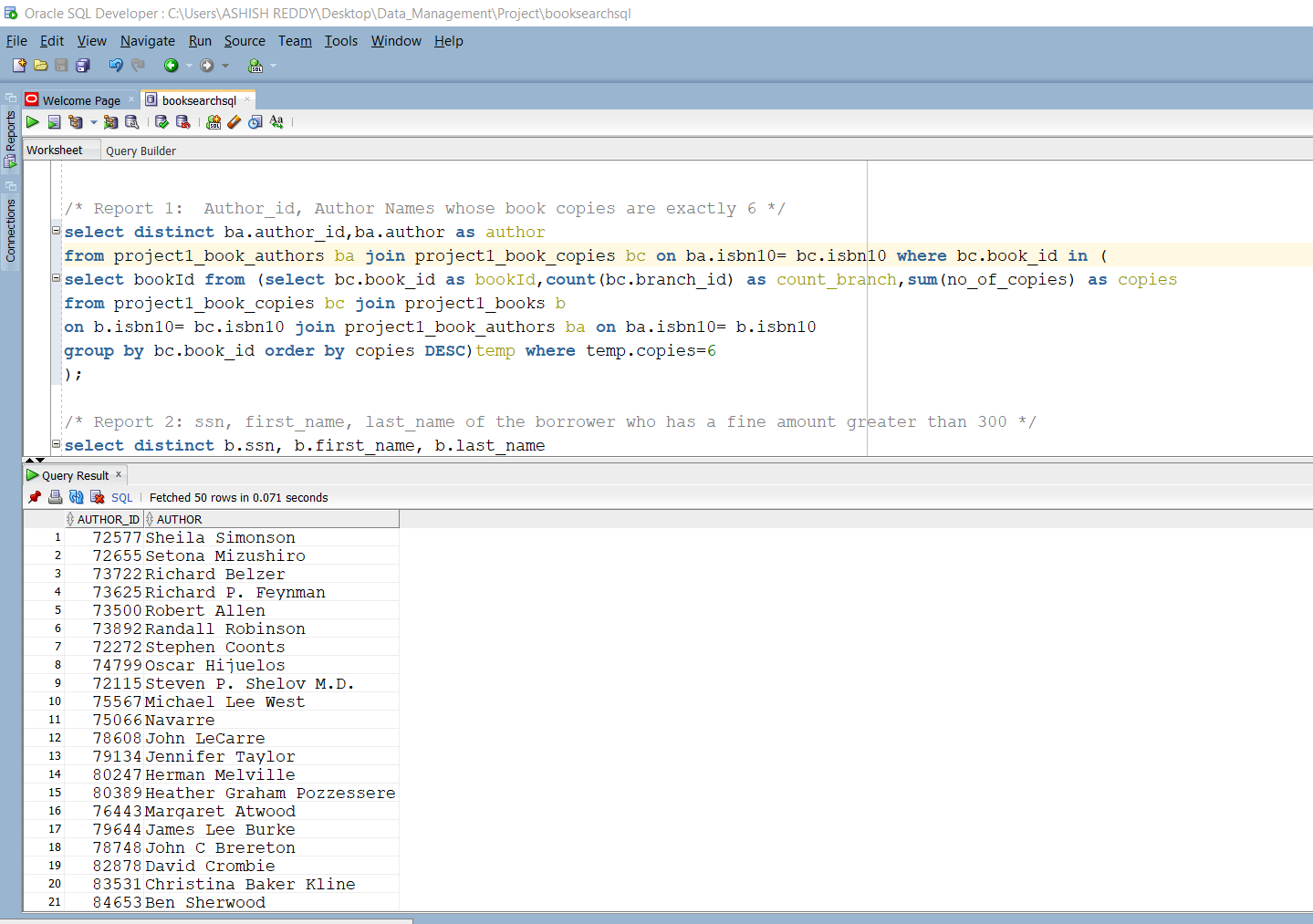
from project1\_book\_copies bc join project1\_books b

on b.isbn10= bc.isbn10 join project1\_book\_authors ba

on ba.isbn10= b.isbn10

group by bc.book\_id

order by copies DESC) temp where temp.copies = 6);



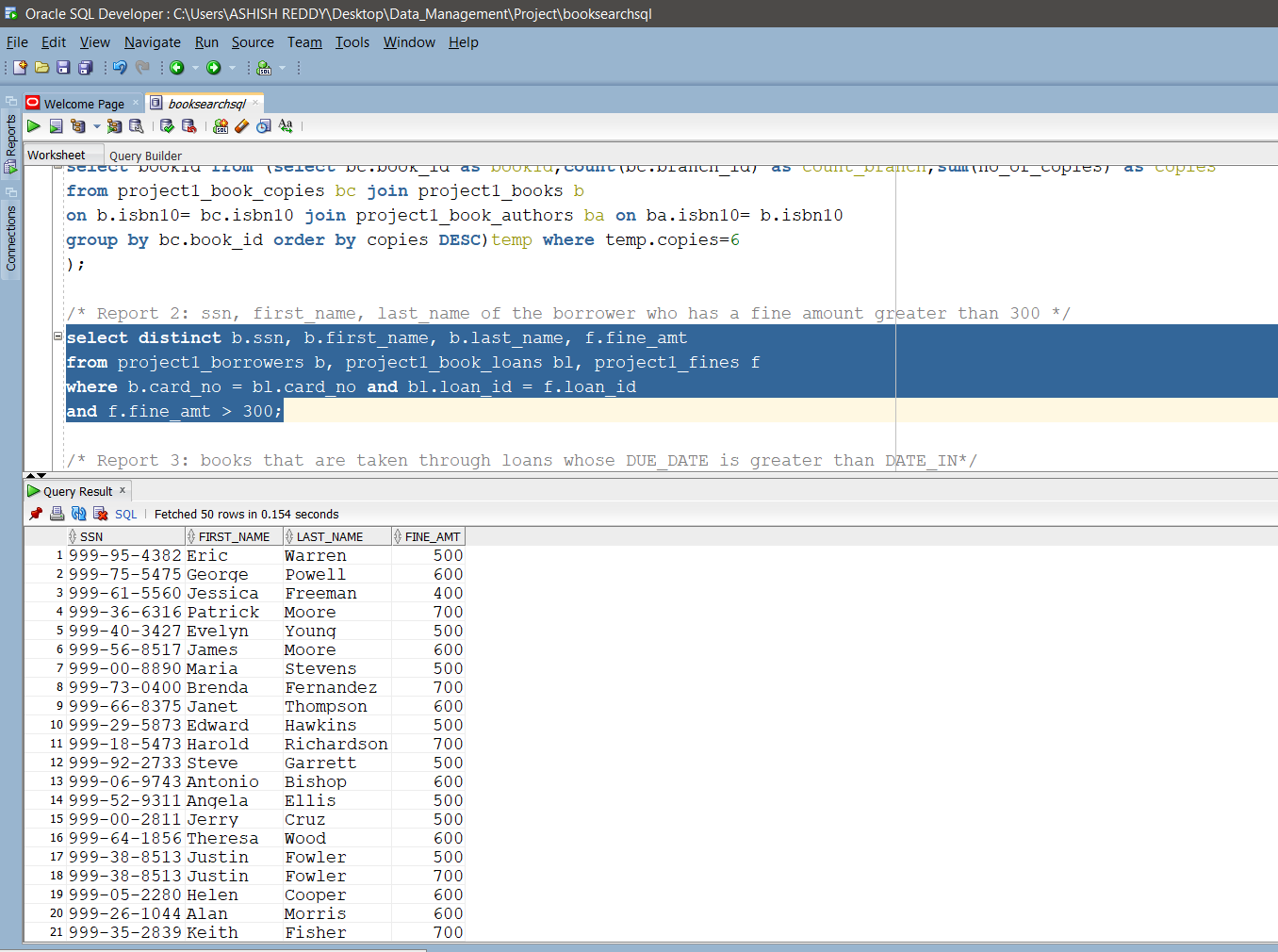
/\* Report 2: ssn, first\_name, last\_name of the borrower who has a fine amount greater than 300 \*/

select distinct b.ssn, b.first\_name, b.last\_name

from project1\_borrowers b, project1\_book\_loans bl, project1\_fines f

where b.card\_no = bl.card\_no and bl.loan\_id = f.loan\_id

and f.fine\_amt > 300;



/\* Report 3: books that are taken through loans whose DUE\_DATE is greater than DATE\_IN\*/

select distinct b.title

from project1\_books b, project1\_book\_copies bc, project1\_book\_loans bl

where b.ISBN10 = bc.ISBN10 and bc.book\_id = bl.book\_id

and bl.Due\_date > bl.DATE\_IN;

