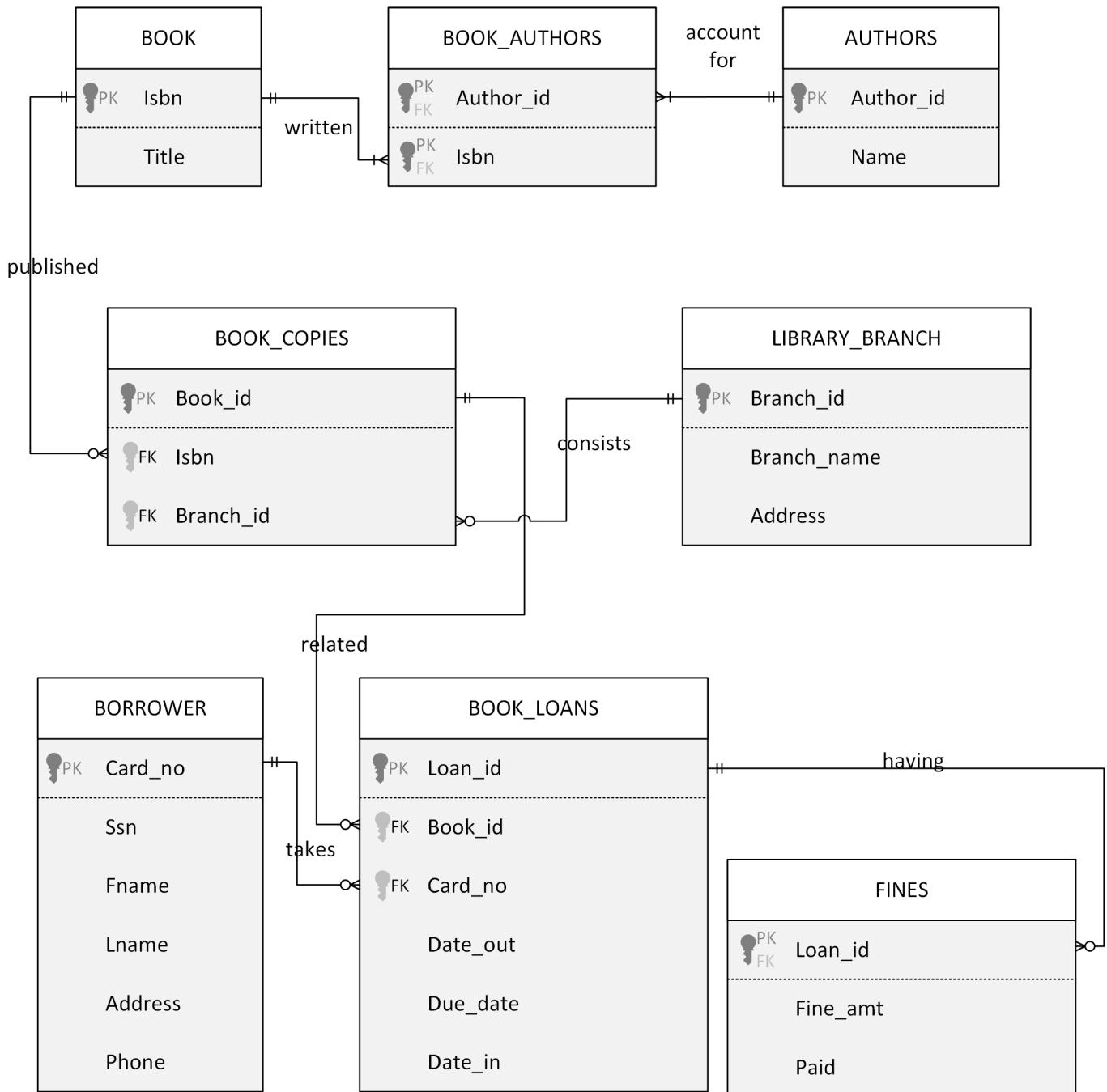


1. Design and DB Architecture:

a. ERD Diagram

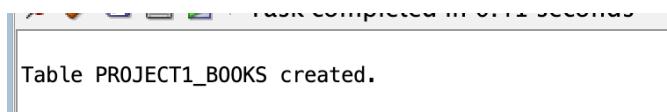


b. Database objects create statements (Tables creation)

Book table creation

```
drop table project1_books;

create table Project1_books (
ISBN VARCHAR2(26) primary key,
Title VARCHAR2(256));
```



A screenshot of the Oracle SQL Developer interface. The code editor window contains the SQL statements for creating the 'Project1_books' table. Below the editor, a message window displays the confirmation: 'Table PROJECT1_BOOKS created.' Above the message window, a status bar shows 'Task completed in 0.12 seconds'.

Table PROJECT1_BOOKS created.

Books_Authors table creation

```
drop table project1_books_authors;

create table project1_books_authors(
Author_ID NUMBER(20) references
project1_authors(author_id),
ISBN VARCHAR(26) references
project1_books(ISBN));
```



A screenshot of the Oracle SQL Developer interface. The code editor window contains the SQL statements for creating the 'Project1_books_authors' table. Below the editor, a message window displays the confirmation: 'Table PROJECT1_BOOKS_AUTHORS created.'

Table PROJECT1_BOOKS_AUTHORS created.

Authors table creation

```
drop table project1_authors;
create table project1_authors (
AUTHOR_ID INT NOT NULL,
AUTHRO VARCHAR2(150));
alter table project1_authors
rename column
AUTHRO to name;

alter table project1_authors add(
constraint Author_pk PRIMARY KEY (AUTHOR_ID));

drop sequence Author_PK_Sq;
CREATE SEQUENCE Author_PK_Sq START WITH 1
increment by 1;
```

Table PROJECT1_AUTHORS created.

Table PROJECT1_AUTHORS altered.

Sequence AUTHOR_PK_SQ created.

Library_Branch table creation

```
drop table project1_Library_branch;

create table project1_Library_branch(
Branch_ID number(10),
Branch_name varchar(25),
address varchar(50));

alter table project1_Library_branch add constraint Branch_id_pk primary key
(Branch_ID);
```

Table PROJECT1_LIBRARY_BRANCH dropped.

Table PROJECT1_LIBRARY_BRANCH created.

Table PROJECT1_LIBRARY_BRANCH altered.

Book_copies table creation

```
drop table project1_book_copies;

create table project1_book_copies (
    Book_id number(10) primary key,
    ISBN VARCHAR(26) references project1_books(ISBN),
    Branch_id number(10) references project1_library_branch(branch_id));

create sequence book_id_sq start with 1 increment by 1;
```

Table PROJECT1_BOOK_COPIES created.

Sequence BOOK_ID_SQ created.

Borrower table creation

```
drop table project1_borrower;

create table project1_borrower(
    card_no number (38),
    ssn varchar2(100),
    fname varchar2(100),
    lname varchar2(100),
    address varchar2(100),
    phone varchar2(100));

alter table project1_borrower add constraint card_no_pk primary key
(card_no);

create sequence card_no_sk start with 100 increment by 1;
```

Table PROJECT1_BORROWER created.

Table PROJECT1_BORROWER altered.

Sequence CARD_NO_SK created.

Book_loans table creation

```
drop table project1_book_loans;

create table project1_book_loans(
loan_id varchar2(250) primary key,
book_id number(10) references project1_book_copies(Book_id),
card_no number (38) references project1_borrower(card_no),
date_out date, due_date date, date_in date);

create sequence loan_id_sq
start with 1 increment by 1;
```

Table PROJECT1_BOOK_LOANS created.

Sequence LOAN_ID_SQ created.

Fines table creation

```
drop table project1_fines;

create table project1_fines(
loan_id varchar2(250) references
project1_book_loans,
fine_amt number(10),
paid varchar2(10));
```

Table PROJECT1_FINES created.

b. Data Load, Normalization, data generation

a. Initial load of files to tables

Books_load

```
select * from project1_books_load;
```

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help, and a date/time stamp of Fri Oct 29 12:53:29 AM. The left sidebar contains sections for Synonyms, Database Links, Public Database Links, Directories, Editions, Java, XML Schemas, XML DB Repository, OLAP Option, Analytic Views, and Scheduler. Below that is a Database Schema Service Connections section with options like All Reports, Analytic View Reports, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, TimesTen Reports, and User Defined Reports. The central workspace has tabs for DBF54.sql, Welcome Page, project1.sql, and project1.spr. The Worksheet tab displays the query: `select * from project1_books_load;`. The Query Result tab shows the output of the query, which is a table with columns: ISBN10, ISBN13, TITLE, AUTHRO, COVER, PUBLISHER, and PAGES. The table contains 50 rows of book data. At the bottom of the interface, there are tabs for Messages - Log, Logging Page, and Statements, along with status indicators for Line 3 Column 1, Insert, Modified, and Unix/Mac: LF.

ISBN10	ISBN13	TITLE	AUTHRO	COVER	PUBLISHER	PAGES
1 0195153448	9780195153446	Classic...	Marc P. ...	http://www.openisbn.com/cover/...	Oxford University Press, USA	844
2 002005018	9780002005012	Clara C...	Richard ...	http://www.openisbn.com/cover/...	HarperFlamingoCanada	414
3 0060973129	9780060973124	Decisio...	Carlo D'...	http://www.openisbn.com/cover/...	HarperPerennial	555
4 0374157065	9780374157067	Flu: Th...	Gina Kolata	http://www.openisbn.com/cover/...	Farrar, Straus And Giroux	256
5 0393045218	9780393045215	The Mum...	Elizabeth...	http://www.openisbn.com/cover/...	W. W. Norton Samp; Company	240
6 0399135782	9780399135781	The Kit...	Amy Tan	http://www.openisbn.com/cover/...	G. P. Putnam's Sons	415
7 0425176428	9780425176429	What If...	Robert C...	http://www.openisbn.com/cover/...	Berkley Trade	416
8 0671870432	9780671870430	Pleadi...	Scott Tu...	http://www.openisbn.com/cover/...	Simon & Schuster Audio	0
9 0679425608	9780679425601	Under T...	David Co...	http://www.openisbn.com/cover/...	Random House	296
10 074322678X	9780743226783	Where Y...	Ann Beattie	http://www.openisbn.com/cover/...	Scribner	208
11 0771074670	9780771074677	Nights ...	David Ad...	http://www.openisbn.com/cover/...	Emblemed Editions	225
12 080652121X	9780806521213	Hitle...	Kensington	http://www.openisbn.com/cover/...	Citadel	293
13 0887841740	9780887841743	The Mid...	Sheila Heti	http://www.openisbn.com/cover/...	House Of Anansi Press	160
14 1552041778	9781552041772	Jane Doe	J. R. Ka...	http://www.openisbn.com/cover/...	Mira Books	0
15 1558746218	9781558746213	A Secon...	Jack Can...	http://www.openisbn.com/cover/...	HCI	328
16 1567407781	9781567407785	The Wit...	Loren D....	http://www.openisbn.com/cover/...	Nova Audio Books	0
17 1575663937	9781575663937	More Cu...	Robert H...	http://www.openisbn.com/cover/...	Kensington	288
18 1881320189	9781881320180	Goodbye...	Julia Ol...	http://www.openisbn.com/cover/...	River City Pub	191
19 0440234743	9780440234746	The Tes...	John Gri...	http://www.openisbn.com/cover/...	Dell	544
20 0440234746	9780440234746	The Tes...	John Gri...	http://www.openisbn.com/cover/...	Dell	575

Book_copies_load

```
select * from project1_book_copies_load;
```

Oracle SQL Developer File Edit View Navigate Run Source Team Tools Window Help Fri Oct 29 12:56:04 AM

DBFS DBFS

SQL Worksheet History

Worksheet Query Builder

```
select * from project1_book_copies_load;
```

Query Result Fetched 50 rows in 0.061 seconds

BOOK_ID	BRANCH_ID	NO_OF_COPIES
1 0195153448	1	1
2 0195153448	2	0
3 0195153448	3	1
4 0195153448	4	1
5 0195153448	5	0
6 0002005018	1	1
7 0002005018	2	1
8 0002005018	3	0
9 0002005018	4	0
10 0002005018	5	1
11 0060973129	1	0
12 0060973129	2	1
13 0060973129	3	0
14 0060973129	4	1
15 0060973129	5	1
16 0374157065	1	1
17 0374157065	2	1
18 0374157065	3	1
19 0374157065	4	1
20 0374157065	5	1
21 0393045218	1	0
22 0393045218	2	1

Borrowers_load

```
select * from project1_borrowers_load;
```

Connections

DBFS DBFS

SQL Worksheet History

Worksheet Query Builder

```
select * from project1_borrowers_load;
select * from project1_library_branch_load;
```

Query Result Fetched 50 rows in 0.073 seconds

ID000001	SSN	FIRST_NAME	LAST_NAME	EMAIL	ADDRESS	CITY	ST
1 ID000001	999-47-3740	Mark	Morgan	mmorgan0@g.co	5677 Coolidge Street	Plano	TX
2 ID000002	999-95-4382	Eric	Warren	ewarren1@ed.gov	9062 Schurz Drive	Dallas	TX
3 ID000003	999-51-5268	Robert	Harper	rharper2@angelfire.com	7786 Sachs Place	Plano	TX
4 ID000004	999-61-3567	Deborah	Franklin	dfranklin@alexa.com	3 Magdalene Terrace	Plano	TX
5 ID000005	999-63-3588	Linda	King	lking4@illinois.edu	41260 Kedzie Terrace	Dallas	TX
6 ID000006	999-26-7141	Judy	Holmes	jholmes5@github.io	6583 Manitowish Hill	Dallas	TX
7 ID000007	999-61-3567	Deborah	Lawrence	dlawrence@chicagotribune.com	2 Hermina Trail	Dallas	TX
8 ID000008	999-25-0492	Adam	Johnston	ajohnston7@is.gd	05671 Norway Maple Court	Dallas	TX
9 ID000009	999-12-4697	Pamela	Carr	pcarr@stumbleupon.com	1848 Fordem Avenue	Richardson	TX
10 ID000010	999-33-7951	Daniel	Fisher	dfisher@liveinternet.ru	400 Cody Lane	Plano	TX
11 ID000011	999-90-7753	Stephen	Fields	sfieldsa@phinn.com	99 Fair Oaks Drive	Plano	TX
12 ID000012	999-53-3736	Walter	Ferguson	wferguson@album.net	814 Sutteridge Point	Dallas	TX
13 ID000013	999-61-6311	Paul	Cook	pcook@blogtalkradio.com	58295 Pierstorff Pass	Richardson	TX
14 ID000014	999-24-6360	Roy	Smith	rsmithdcmonitor.com	563 Fairview Park	Dallas	TX
15 ID000015	999-83-8412	Michelle	Gardner	mgardnerew@wordpress.org	28309 Paget Court	Plano	TX
16 ID000016	999-75-5475	George	Powell	gpowellf@scribd.com	31 Clove Crossing	Dallas	TX
17 ID000017	999-61-5822	Nicholas	Hudson	nhudson@uol.com.br	29 Esker Trail	Dallas	TX
18 ID000018	999-82-7179	Linda	Henderson	lhendersoh@mayoclinic.com	117 Katie Center	Dallas	TX
19 ID000019	999-96-1826	Gregory	Fowler	gfowler1@foxnews.com	42454 Hermina Plaza	Dallas	TX
20 ID000020	999-52-7080	Carol	Sanders	csandersj@usda.gov	99600 Moland Way	Dallas	TX
21 ID000021	999-61-5170	Mark	Hamilton	mhamilton@deliciousdays.com	76425 Dixon Hill	Dallas	TX
22 ID000022	999-71-1123	Anthony	Brown	abrown1@rakuten.co.jp	8691 Walton Terrace	Dallas	TX

Library_branch_load

```
select * from project1_library_branch_load;
```

The screenshot shows the Oracle SQL Developer interface. On the left, there's a tree view of database objects under 'Connections' and 'Reports'. The main area has tabs for 'DBFS4.sql', 'Welcome Page', 'project1 (2).sql', and 'project1.sql'. The 'project1.sql' tab is active, showing a 'Worksheet' tab with the SQL query: 'select * from project1_library_branch_load;'. Below it is a 'Query Result' tab displaying a table with 5 rows of data:

BRANCH_ID	BRANCH_NAME	ADDRESS
1	1 Oak Lawn	4100 Cedar Springs Road, 75219
2	2 Lakewood	6121 Worth Street, 75214
3	3 Grauwylter Park	2146 Gilford Street, 75235
4	4 Highland Hills	3624 Simpson Stuart Road, 75241
5	5 Audelia Road	10045 Audelia Road, 75238

b. SQL STATEMENTS

Normalizing Data

Populating Books Table:

```
insert into project1_books (title, ISBN)
select DISTINCT title, ISBN10 from
project1_books_load;
commit;
```

```
select * from project1_books;
```

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree is expanded to show various database objects like Synonyms, Database Links, Directories, Editions, Java, XML Schemas, XML DB Repository, OLAP Option, Analytic Views, Scheduler, RDF Semantic Graph, Recycle Bin, and Other Users. Below that is the Database Schema Service Connections section with reports like All Reports, Analytic View Reports, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, TimesTen Reports, and User Defined Reports.

The main workspace has tabs for DBFS4.sql, Welcome Page, project1 (2).sql, and project1.sql. The project1.sql tab is active, showing a SQL Worksheet with the following code:

```
--Populating books table--
insert into project1_books (title, ISBN)
select DISTINCT title, ISBN10 from project1_books_load;
commit;

select * from project1_books;
```

Below the worksheet is a Script Output window showing the results of the query:

SQL : Fetched 50 rows in 0.045 seconds

ISBN	TITLE
1	1573245216 The Giving Heart: Unlocking The Transformative Power Of Generosity In Your Life
2	0670891576 In The Heart Of The Sea: The Tragedy Of The Whaleship Essex
3	0340542667 Canon Bang-bang
4	0151006725 Slammerkin: A Loose Dress, A Loose Woman
5	2070756319 Le Dictateur Et Le Hamac
6	0446674346 Cloud Mountain
7	002026478X Age Of Innocence (Movie Tie-in)
8	034541103X A Reporter's Life
9	038000772X I'm OK-You're OK
10	0553208799 Chancellor Manuscript
11	0679401032 Uh-Oh: Some Observations From Both Sides Of The Refrigerator Door
12	0394571029 All I Really Need To Know I Learned In Kindergarten: Uncommon Thoughts On Common Things
13	002542730X Politically Correct Bedtime Stories: Modern Tales For Our Life & Times
14	089190672X Circus
15	3423202947 Hunde Von Riga: Roman
16	059048575X 101 Telephone Jokes
17	0679401032 Uh-Oh: Some Observations From Both Sides Of The Refrigerator Door

At the bottom, there are tabs for Messages - Log, messages, Logging Page, and Statements. The status bar shows Line 27 Column 1, Insert, Modified!, Unix/Mac: LF.

Populating Library_branch Table:

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

select * from project1_library_branch;
```

```

Connections DBFS4.sql Welcome Page project1 (2).sql project1.sql
SQL Worksheet History
Worksheet Query Builder
alter table project1_Library_branch add constraint Branch_id_pk primary key (Branch_ID);

--populating library branch table

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

select * from project1_library_branch;

--creating borrowers table

drop table project1_borrower;

create table project1_borrower(
card_no number (38);

```

Script Output | Query Result | All Rows Fetched: 5 in 0.045 seconds

BRANCH_ID	BRANCH_NAME	ADDRESS
1	1 Oak Lawn	4100 Cedar Springs Road, 75219
2	2 Lakewood	6121 Worth Street, 75214
3	3 Grauwylde Park	2146 Gilford Street, 75235
4	4 Highland Hills	3624 Simpson Stuart Road, 75241
5	5 Audelia Road	10045 Audelia Road, 75238

Messages - Log | Messages | Logging Page | Statements | Line 231 Column 1 | Insert | Modified | Unix/Mac: LF

Populating Borrower Table:

```

insert into project1_borrower(card_no,ssn,fname,lname,address,phone)
select card_no_sk.nextval,ssn(first_name,last_name,address,phone from
project1_borrowers_load;

commit;

```

```

Connections DBFS4.sql Welcome Page project1 (2).sql project1.sql
SQL Worksheet History
Worksheet Query Builder
insert into project1_borrower(card_no,ssn,fname,lname,address,phone)
select card_no_sk.nextval,ssn(first_name,last_name,address,phone from
project1_borrowers_load;

commit;

select * from project1_borrower;

commit;

```

Script Output | Query Result | Fetched 50 rows in 0.062 seconds

CARD_NO	SSN	FNAME	LNAME	ADDRESS	PHONE
1	417 999-37-2302	Linda	Richards	046 Harper Center	(214) 998-2489
2	418 999-78-8042	Albert	Cunningham	414 Summer Ridge Trail	(972) 716-3924
3	419 999-16-9228	Debra	Holmes	22 Bluejay Road	(972) 954-8773
4	420 999-33-5177	Jason	Davis	82931 Cardinal Junction	(214) 969-9787
5	421 999-38-8513	Justin	Fowler	1950 Leroy Way	(972) 915-2410
6	422 999-58-4650	Catherine	Harris	4091 Doe Crossing Alley	(469) 968-7984
7	423 999-94-6409	Gary	Reyes	06 Shelley Parkway	(972) 985-2014
8	424 999-37-2293	Clarence	Harvey	637 Blackbird Crossing	(972) 735-8046
9	425 999-26-3470	Kathleen	Lynch	0 Northview Center	(469) 858-9821
10	426 999-87-1283	Angela	Myers	07 Jenifer Point	(214) 996-3258
11	427 999-30-0056	Paula	Roberts	2025 Annamarck Court	(214) 919-1512
12	428 999-25-1300	Marie	Hudson	35523 Fulton Point	(972) 955-0047
13	429 999-83-3012	Victor	George	16 Butterfield Place	(469) 982-0024
14	430 999-53-7223	Judith	Lane	59463 Havey Center	(469) 962-1701
15	431 999-13-1583	Steven	Carpenter	64 Rutledge Court	(972) 759-2760

Messages - Log | Messages | Logging Page | Statements | Line 259 Column 1 | Insert | Modified | Unix/Mac: LF

Populating Book_Copies Table:

```
DECLARE
rep_cnt NUMBER;
CURSOR c_trx
IS
SELECT
    book_id, branch_id, no_of_copies
  FROM project1_book_copies_load;
BEGIN
FOR r_trx IN c_trx
LOOP

rep_cnt := r_trx.no_of_copies;
LOOP
IF rep_cnt <= 0
THEN
    EXIT;
END IF;
insert into project1_book_copies (book_id, ISBN, branch_id) values
(book_id_sq.nextval, r_trx.book_id, r_trx.branch_id) ;
rep_cnt := rep_cnt-1;
END LOOP;
END LOOP;
END;
/
commit;

select * from project1_book_copies;
```

The screenshot shows the Oracle SQL Developer interface during the execution of a PL/SQL script. The top navigation bar includes tabs for 'DBFS4.sql', 'Welcome Page', 'project1 (2).sql', and 'project1.sql'. The main workspace displays the script code. Below the workspace is a 'Script Output' tab showing the results of the executed query. The results are presented in a table with columns: BOOK_ID, ISBN, and BRANC... (partially visible).

BOOK_ID	ISBN	BRANC...
1	951 0553278746	3
2	952 0553278746	4
3	953 0553278746	5
4	954 0570856045	2
5	955 0570856045	3
6	956 0670856045	4
7	957 0570856045	5
8	958 0872200760	1
9	959 0872200760	3
10	960 0872200760	4
11	961 0872200760	5
12	962 0937858994	1
13	963 0937858994	2
14	964 0937858994	3
15	965 0937858994	4

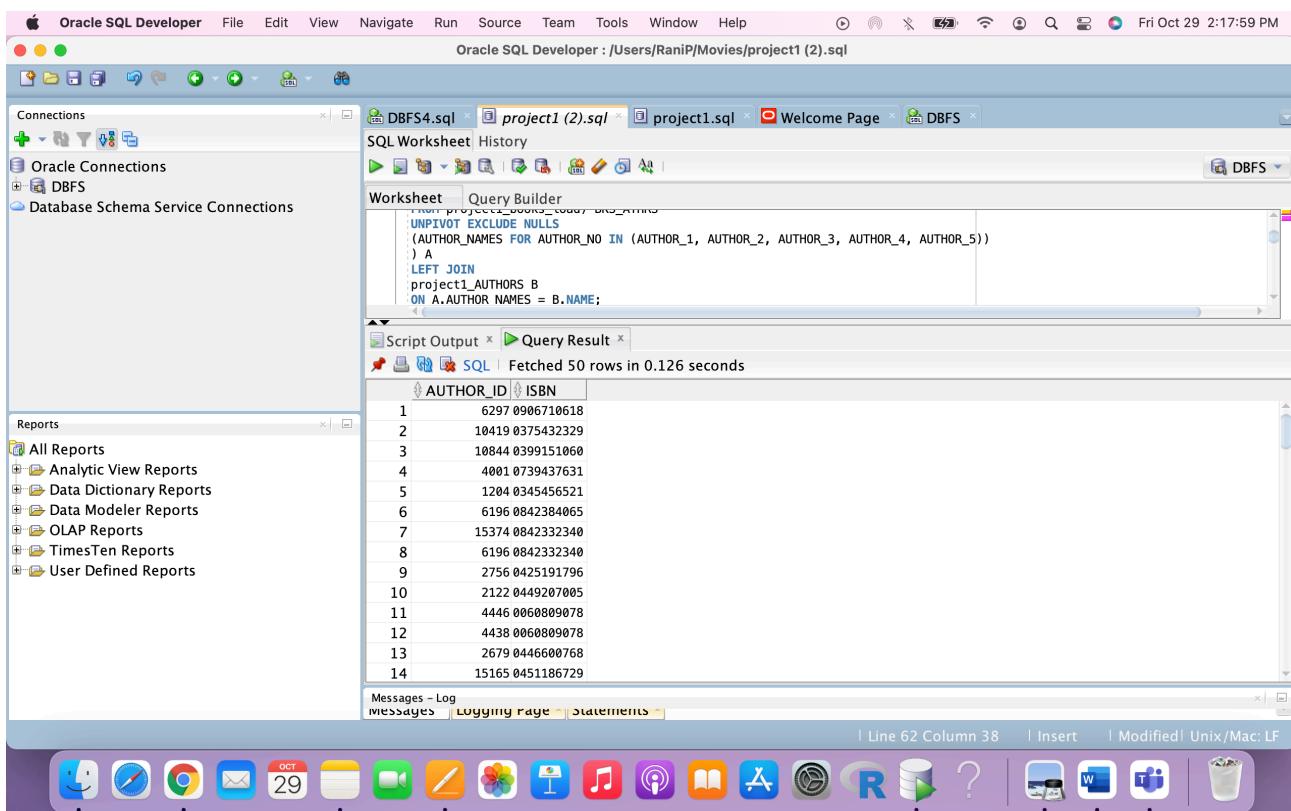
At the bottom of the interface, there are tabs for 'Messages - Log', 'Messages', 'Logging Page', and 'Statements'. A status bar at the very bottom indicates 'Line 290 Column 1 | Insert | Modified | Unix/Mac: LF'.

Populating Book_Authors Table:

```
INSERT INTO project1_books_authors
SELECT AUTHOR_ID, ISBN10 AS ISBN
FROM
(
  SELECT ISBN10, AUTHOR_NAMES
  FROM (SELECT ISBN10, REGEXP_SUBSTR(AUTHRo, '[^,]+', 1, 1) AS AUTHOR_1,
              REGEXP_SUBSTR(AUTHRo, '[^,]+', 1, 2) AS AUTHOR_2,
              REGEXP_SUBSTR(AUTHRo, '[^,]+', 1, 3) AS AUTHOR_3,
              REGEXP_SUBSTR(AUTHRo, '[^,]+', 1, 4) AS AUTHOR_4,
              REGEXP_SUBSTR(AUTHRo, '[^,]+', 1, 5) AS AUTHOR_5
  FROM project1_books_load) BKS_ATHRS
  UNPIVOT EXCLUDE NULLS
  (AUTHOR_NAMES FOR AUTHOR_NO IN (AUTHOR_1, AUTHOR_2, AUTHOR_3, AUTHOR_4,
  AUTHOR_5))
) A
LEFT JOIN
project1_AUTHORS B
ON A.AUTHOR_NAMES = B.NAME;

COMMIT;

select * from project1_books_authors;
```



Populating Authors Table:

```
insert into project1_authors(AUTHOR_ID, name)
select Author_PK_Sq.nextval, part_1
from temp_auth_name;
COMMIT;

select * from project1_authors;
```

The screenshot shows the Oracle SQL Developer interface. On the left, there's a sidebar with 'Connections' (DBFS selected), 'Reports' (All Reports, Analytic View Reports, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, TimesTen Reports, User Defined Reports), and a 'Script Output' tab. The main area has tabs for 'DBFS4.sql', 'project1 (2).sql', 'project1.sql', and 'Welcome Page'. The 'project1.sql' tab is active, displaying the SQL code from the previous block. Below it is a 'Query Result' tab showing the output of the query:

AUTHOR_ID	NAME
1	1 (None)
2	2 (none)
3	3 A. A. Milne
4	4 A. E. Van Vogt
5	5 A. First Book
6	6 A. Oswald
7	7 A. Redemptorist Pastoral Publication
8	8 A. Lelia Bundles
9	9 A. A. Attanasio
10	10 A. A. Fair
11	11 A. A. Milne
12	12 A. Alvarez
13	13 A. B. C. Whipple
14	14 A. B. Guthrie Jr.
15	15 A. C. Bhaktivedanta Swami Prabhupada
16	16 A. C. Bhaktivedanta Swami Prabhupāda
17	17 A. C. Crispin

At the bottom, there are tabs for 'Messages - Log', 'statements', and 'Logging Page'. Status information at the bottom right includes 'Line 209 Column 32', 'Insert', 'Modified', and 'Unix/Mac: LF'.

Creating and populating the temporary table to populate Authors Table:

```
create table t1(
a varchar2(200));
create table t2(
a varchar2(200));
create table t3(
a varchar2(200));
create table t4(
a varchar2(200));
create table t5(
a varchar2(200));
drop table t6;
drop table t7;
delete from t1;
delete from t2;
delete from t3;
delete from t4;
delete from t5;

insert into t1
SELECT distinct
trim(REGEXP_SUBSTR(authro, '[^,]+', 1, 1) ) a
FROM PROJECT1_BOOKS_LOAD where REGEXP_SUBSTR(authro, '[^,]+', 1, 1) is
not null
order by a;
insert into t2
SELECT distinct
trim(REGEXP_SUBSTR(authro, '[^,]+', 1, 2)) a
FROM PROJECT1_BOOKS_LOAD where REGEXP_SUBSTR(authro, '[^,]+', 1, 2) is
not null
order by a;
insert into t3
SELECT distinct
trim(REGEXP_SUBSTR(authro, '[^,]+', 1, 3)) a
FROM PROJECT1_BOOKS_LOAD where REGEXP_SUBSTR(authro, '[^,]+', 1, 3) is
not null
order by a;
insert into t4
SELECT distinct
trim(REGEXP_SUBSTR(authro, '[^,]+', 1, 4)) a
FROM PROJECT1_BOOKS_LOAD where REGEXP_SUBSTR(authro, '[^,]+', 1, 4) is
not null
order by a;
insert into t5
SELECT distinct
trim(REGEXP_SUBSTR(authro, '[^,]+', 1, 5)) a
FROM PROJECT1_BOOKS_LOAD where REGEXP_SUBSTR(authro, '[^,]+', 1, 5) is
not null
order by a;

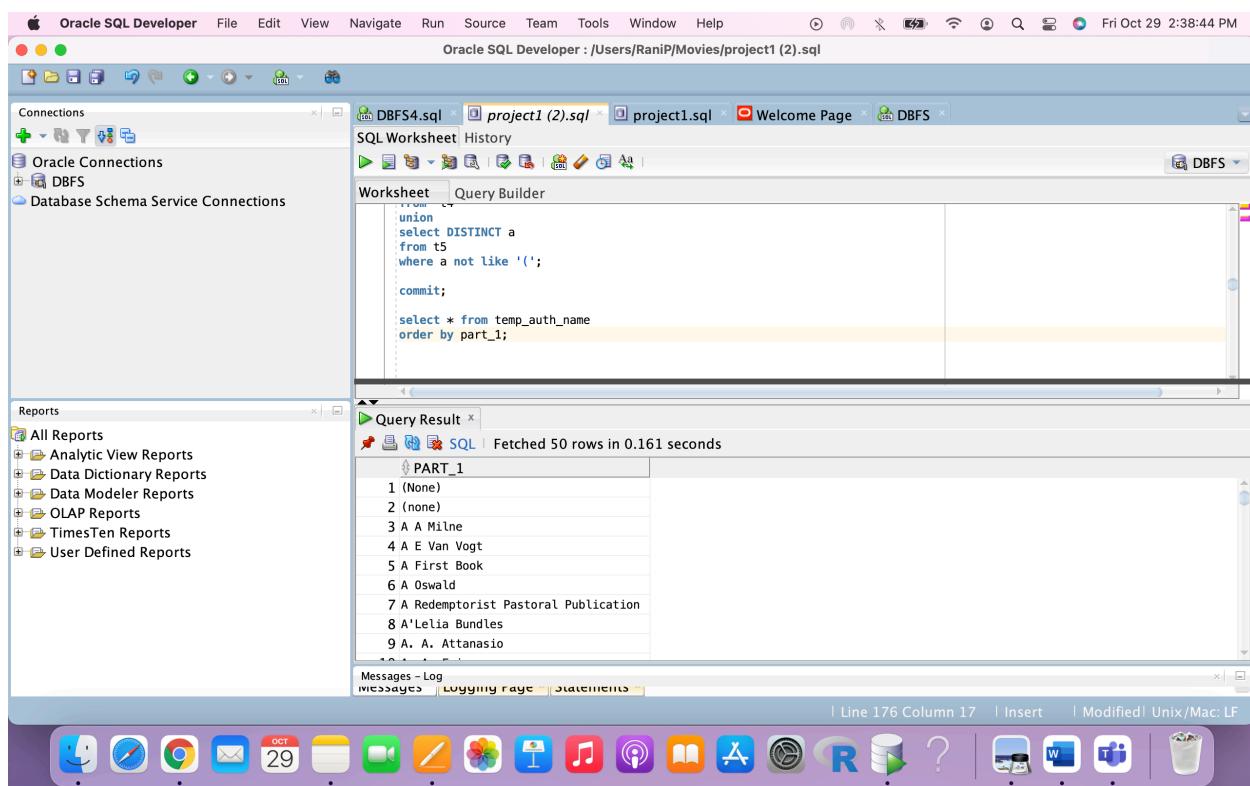
drop table temp_auth_name;

create table temp_auth_name(
part_1 VARCHAR2(200));
```

```
delete from temp_auth_name;
insert into temp_auth_name(part_1)
select DISTINCT a
from t1
union
select DISTINCT a
from t2
union
select DISTINCT a
from t3
union
select DISTINCT a
from t4
union
select DISTINCT a
from t5
where a not like '(';

commit;

select * from temp_auth_name
order by part_1;
```



Generating SQL with conditions:

- (1) Exactly 400 books check-outs for exactly 200 different borrowers and exactly 100 different books. Same borrower should not check out same book more than once

Populating Book_Loans_Table:

```
insert into project1_book_loans(loan_id,book_id,card_no,date_out,due_date,date_in)
select
loan_id_sq.nextval,book_id,card_no,date_out,
date_out + 10 as due_date,
case when rownum <=50
then date_out+10 + round(dbms_random.value(1,15),0)
else date_out+10 - round(dbms_random.value(0,10),0)
end
from (select
a.card_no as card_no,
b.book_id as book_id,
dense_rank() over (order by card_no) as ar,
row_number() over (partition by card_no order by card_no,book_id) as abr,
sysdate - round(dbms_random.value(1,30),0) as date_out
from (
select * from (select card_no from project1_borrower
order by dbms_random.random) where rownum <= 200)a,
(select * from (select book_id from project1_book_copies
order by dbms_random.random)
where rownum <=100)b
order by dbms_random.random)
where (ar)=abr-1
OR (ar)=abr
OR (ar-100)=abr
OR (ar-100)=abr -1
OR (ar = 200 and abr=1);

commit;

select * from project1_book_loans;

select count(distinct card_no) from project1_book_loans;
commit;
```

DBFS4.sql Welcome Page project1 (2).sql project1.sql

Worksheet Query Builder

```

commit;

select * from project1_book_loans;

select count(distinct card_no) from project1_book_loans;
commit;

-----creating fines
drop table project1 fines;

```

Script Output x | Query Result x | SQL | Fetched 50 rows in 0.517 seconds

LOAN_ID	BOOK_ID	CARD_NO	DATE_OUT	DEU_DATE	DATE_IN
1 1	22683	290	23-OCT-21	02-NOV-21	13-NOV-21
2 2	23634	291	10-OCT-21	20-OCT-21	23-OCT-21
3 3	21729	825	21-OCT-21	31-OCT-21	06-NOV-21
4 4	8139	733	28-OCT-21	07-NOV-21	19-NOV-21
5 5	5769	145	02-OCT-21	12-OCT-21	20-OCT-21
6 6	69226	502	04-OCT-21	14-OCT-21	24-OCT-21
7 7	89716	1090	05-OCT-21	15-OCT-21	26-OCT-21
8 8	52149	424	19-OCT-21	29-OCT-21	11-NOV-21
9 9	47439	358	08-OCT-21	18-OCT-21	24-OCT-21
10 10	27521	292	29-SEP-21	09-OCT-21	11-OCT-21
11 11	60282	456	20-OCT-21	30-OCT-21	01-NOV-21
12 12	18401	197	16-OCT-21	26-OCT-21	27-OCT-21
13 13	43869	914	03-OCT-21	13-OCT-21	23-OCT-21
14 14	93333	703	23-OCT-21	02-NOV-21	11-NOV-21
15 15	63353	489	23-OCT-21	02-NOV-21	10-NOV-21
16 16	16743	791	16-OCT-21	26-OCT-21	04-NOV-21

Messages - Log | Logging Page | Statements

Line 368 Column 35 | Insert | Modified | Unix/Mac: LF

DBFS4.sql Welcome Page project1 (2).sql project1.sql

Worksheet Query Builder

```

-----from -----
select * from (select card_no from project1_borrower
order by dbms_random.random) where rownum <= 200)a,
(select * from (select book_id from project1_book_copies
order by dbms_random.random)
where rownum <=100)b
order by dbms_random.random)
where (ar)=abr-1
OR (ar)=abr
OR (ar-100)=abr
OR (ar-100)=abr -1
OR (ar = 200 and abr=1);

commit;

select * from project1_book_loans;

select count(distinct card_no) from project1_book_loans;
commit;

-----creating fines
-----

```

Script Output x | Query Result x | SQL | All Rows Fetched: 1 in 0.042 seconds

COUNT(DISTINCTCARD_NO)
1
200

Messages - Log | Logging Page | Statements

Line 370 Column 58 | Insert | Modified | Unix/Mac: LF

2. Exactly 50 fines records for 50 different borrowers. Fines should be generated by books checked back in late.

Populating Fines Table:

```
insert into project1_fines(loan_id, fine_amt)
( select loan_id, (trunc(Date_in) - trunc(Due_date)) * 10 from
project1_book_loans
where (trunc(Due_date) - trunc(Date_in)) < 0);

commit;

select * from project1_fines;
```

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database schema with nodes like Synonyms, Public Synonyms, Database Links, etc. The central workspace has four tabs open: DBFS4.sql, Welcome Page, project1 (2).sql, and project1.sql. The project1.sql tab is active, showing a worksheet with the following SQL code:

```
-- creating fines
drop table project1_fines;
create table project1_fines(
loan_id varchar2(250) references project1_book_loans,
fine_amt number(10),
paid varchar2(10));
-- populating fines
delete project1_fines;
insert into project1_fines(loan_id, fine_amt)
```

Below the worksheet is a "Script Output" tab showing the execution results:

All Rows Fetched: 50 in 0.597 seconds

LOAN_ID	FINE_AMT	PAID
2 2	30 (null)	
3 3	60 (null)	
4 4	120 (null)	
5 5	80 (null)	
6 6	100 (null)	
7 7	110 (null)	
8 8	130 (null)	
9 9	60 (null)	
10 10	20 (null)	
11 11	20 (null)	
12 12	10 (null)	
13 13	100 (null)	
14 14	90 (null)	

The bottom status bar indicates: Line 388 Column 1 | Insert | Modified! Unix/Mac: LF

C. Book Search and Availability:

```
SELECT ST.ISBN,ST.title,ST.name as Author,br.branch_name, 'Available' as Availability_Status
from
(Select BA.ISBN
, bo.title
, au.name
from project1_books_authors BA
inner join project1_books BO on ba.isbn = bo.isbn
inner join project1_authors AU on au.author_id = ba.author_id
where bo.title like '%will%'

union

Select BA.ISBN
, bo.title
, au.name as Author
from project1_books_authors BA
inner join project1_books BO on ba.isbn = bo.isbn
inner join project1_authors AU on au.author_id = ba.author_id
where au.name like '%will%'

union

Select BA.ISBN
, bo.title
, au.name as Author
from project1_books_authors BA
inner join project1_books BO on ba.isbn = bo.isbn
inner join project1_authors AU on au.author_id = ba.author_id
where BA.ISBN like '%will%' ) ST
inner join project1_BOOK_COPIES BC on bc.isbn = ST.isbn
inner join project1_library_branch BR on br.branch_id = bc.branch_id
where br.branch_id = 3;
```

Search 1: Using '%will%'

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays a tree view of database schemas: BOWLERS_TMP, ORDERBL_TMP, PROJECT1_AUTHORS, PROJECT1_BOOK_COPIES, PROJECT1_BOOK_COPIES_LOAD, PROJECT1_BOOK_LOANS, PROJECT1_BOOKS, and PROJECT1_BOOKS_AUTHORS. The central workspace contains a query builder window with the following SQL code:

```

SELECT ST.ISBN,ST.title,ST.name as Author,br.branch_name, 'Available' as Availability_Status
from
(SELECT BA.ISBN
, bo.title
, au.name
from project1_books_authors BA
inner join project1_books BO on ba.isbn = bo.isbn
inner join project1_authors AU on au.author_id = ba.author_id
where bo.title like '%will%'

union

Select BA.ISBN
, bo.title
, au.name as Author
from project1_books_authors BA
inner join project1_books BO on ba.isbn = bo.isbn
inner join project1_authors AU on au.author_id = ba.author_id
where au.name like '%will%'

union

Select BA.ISBN
, bo.title
, au.name as Author
from project1_books_authors BA
inner join project1_books BO on ba.isbn = bo.isbn
inner join project1_authors AU on au.author_id = ba.author_id
where BA.ISBN like '%will%' ) ST
inner join project1_BOOK_COPIES BC on bc.isbn = ST.isbn
inner join project1_library_branch BR on br.branch_id = bc.branch_id
where br.branch_id = 3

```

The bottom pane shows the results of the query, titled "Query Result". The results table has columns: ISBN, TITLE, AUTHOR, BRANCH_NAME, and AVAILABILITY_STATUS. The data is as follows:

ISBN	TITLE	AUTHOR	BRANCH_NAME	AVAILABILITY_STATUS
1 931520038	The Mount	Carol Emshwiller	Grauwyler Park	Available
2 0821767240	An Unwilling Bride (Zebra Historical Romance)	Jo Beverley	Grauwyler Park	Available
3 1569470170	Outsider In Amsterdam (Grijpstra & De Gier Mystery)	Janwillem Van De Wetering	Grauwyler Park	Available
4 3442354722	Die Zwillingsschwester: Roman	Dagmar Roth	Grauwyler Park	Available
5 3442354722	Die Zwillingsschwester: Roman	Lisa Scott	Grauwyler Park	Available
6 1930603401	Fondue (Quick & Easy Series) (Quick & Easy (Silverback))	Marlisa Szwilus Dr	Grauwyler Park	Available
7 3499228637	Ticket Nach Tokio	Janwillem Van De Wetering	Grauwyler Park	Available
8 0471213020	Mastering Jakarta Struts	James Goodwill	Grauwyler Park	Available

Search 2: Using '%uni%

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays a tree view of database schemas: BOWLEI, ORDER, PROJECT, AUT, NAM, PROJECT, BOOK, ISBN, BRAI, PROJECT, PROJECT, PROJECT, ISBN, TITL, PROJECT, AUT, ISBN, and PROJECT. The central workspace contains a query builder window with the following SQL code:

```

SELECT ST.ISBN,ST.title,ST.name as Author,br.branch_name, 'Available' as Availability_Status
from
(SELECT BA.ISBN
, bo.title
, au.name
from project1_books_authors BA
inner join project1_books BO on ba.isbn = bo.isbn
inner join project1_authors AU on au.author_id = ba.author_id
where bo.title like '%uni%'

union

Select BA.ISBN
, bo.title
, au.name as Author
from project1_books_authors BA
inner join project1_books BO on ba.isbn = bo.isbn
inner join project1_authors AU on au.author_id = ba.author_id
where au.name like '%uni%'

union

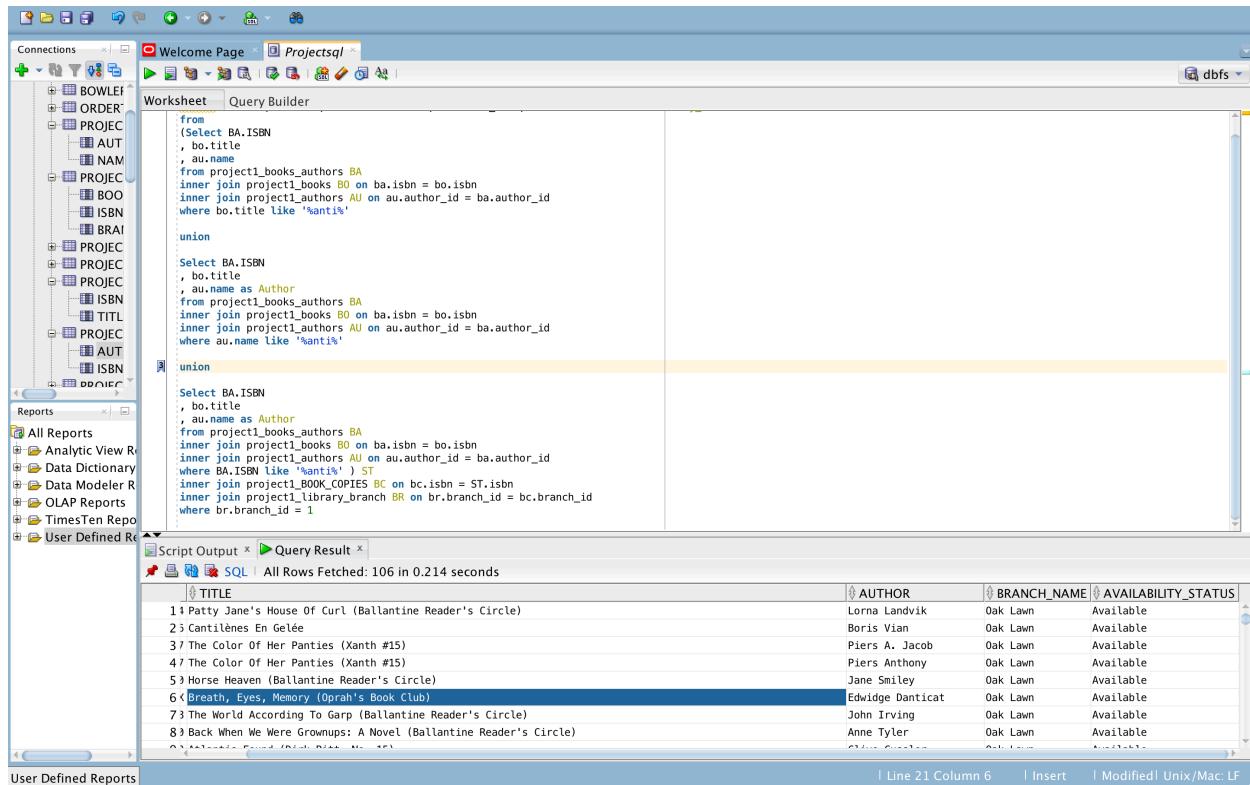
Select BA.ISBN
, bo.title
, au.name as Author
from project1_books_authors BA
inner join project1_books BO on ba.isbn = bo.isbn
inner join project1_authors AU on au.author_id = ba.author_id
where BA.ISBN like '%uni%' ) ST
inner join project1_BOOK_COPIES BC on bc.isbn = ST.isbn
inner join project1_library_branch BR on br.branch_id = bc.branch_id
where br.branch_id = 2

```

The bottom pane shows the results of the query, titled "Query Result". The results table has columns: ISBN, TITLE, AUTHOR, BRANCH_NAME, and AVAILABILITY_STATUS. The data is as follows:

ISBN	TITLE	AUTHOR	BRANCH_NAME	AVAILABILITY_STATUS
1 3446202218	Kreutzersonate: Eine Liebesgeschichte	Helga Van Beuningen	Lakewood	Available
2 0819154946	The Community In America	Roland Warren	Lakewood	Available
3 0819215635	Listening Hearts: Discerning Call In Community	Joseph P. Gill	Lakewood	Available
4 0819215635	Listening Hearts: Discerning Call In Community	R. Taylor McLean	Lakewood	Available
5 0819215635	Listening Hearts: Discerning Call In Community	Suzanne G. Farnham	Lakewood	Available
6 0395710901	Punished By Rewards: The Trouble With Gold Stars, Incentive Plans, A's, Praise And Other Bribes	Alfie Kohn	Lakewood	Available
7 0385483503	Arranged Marriage: Stories	Chitra Banerjee Divakaruni	Lakewood	Available
8 0385497280	The Unknown Errors Of Our Lives: Stories	Chitra Banerjee Divakaruni	Lakewood	Available

Search 3: Using '%anti%'



The screenshot shows the Oracle SQL Developer interface. The left sidebar displays a tree view of database objects under 'Connections' and 'PROJECT'. The main workspace is titled 'Worksheet' and contains a complex SQL query:

```

from
(Select BA.ISBN
, bo.title
, au.name
from project1_books_authors BA
inner join project1_books BO on ba.isbn = bo.isbn
inner join project1_authors AU on au.author_id = ba.author_id
where bo.title like '%anti%'

union

Select BA.ISBN
, bo.title
, au.name as Author
from project1_books_authors BA
inner join project1_books BO on ba.isbn = bo.isbn
inner join project1_authors AU on au.author_id = ba.author_id
where au.name like '%anti%'

union

Select BA.ISBN
, bo.title
, au.name as Author
from project1_books_authors BA
inner join project1_books BO on ba.isbn = bo.isbn
inner join project1_authors AU on au.author_id = ba.author_id
where BA.ISBN like '%anti%' ) ST
inner join project1_BOOK_COPIES BC on bc.isbn = ST.isbn
inner join project1_library_branch BR on br.branch_id = bc.branch_id
where br.branch_id = 1

```

The 'Script Output' tab shows the results of the query, which includes a list of book titles and a table of library branch availability:

TITLE	AUTHOR	BRANCH_NAME	AVAILABILITY_STATUS
11 Patty Jane's House Of Curl (Ballantine Reader's Circle)	Lorna Landvik	Oak Lawn	Available
25 Cantilènes En Gelée	Boris Vian	Oak Lawn	Available
37 The Color Of Her Panties (Xanth #15)	Piers A. Jacob	Oak Lawn	Available
47 The Color Of Her Panties (Xanth #15)	Piers Anthony	Oak Lawn	Available
53 Horse Heaven (Ballantine Reader's Circle)	Jane Smiley	Oak Lawn	Available
64 Breath, Eyes, Memory (Oprah's Book Club)	Edwidge Danticat	Oak Lawn	Available
73 The World According To Garp (Ballantine Reader's Circle)	John Irving	Oak Lawn	Available
83 Back When We Were Grownups: A Novel (Ballantine Reader's Circle)	Anne Tyler	Oak Lawn	Available

D. Reports:

Report 1 : Report of total publications count by each author

```

select a.author_id,a.name,count(b.isbn) as copies
from project1_authors a join project1_books_authors b on
a.author_id=b.author_id join project1_books bk on b.isbn=bk.isbn
group by a.author_id,a.name;

```

The screenshot shows the Oracle SQL Developer interface. On the left, there's a 'Connections' sidebar with 'Oracle Connections' and 'Database Schema Service Connections'. Below it is a 'Reports' sidebar with 'All Reports' expanded, showing categories like 'Analytic View Reports', 'Data Dictionary Reports', 'Data Modeler Reports', 'OLAP Reports', 'TimesTen Reports', and 'User Defined Reports'. The main workspace has tabs for 'DBFS4.sql', 'project1 (2).sql', 'project1.sql', 'Welcome Page', and 'DBFS'. The 'DBFS' tab is active. It contains a 'Worksheet' tab with the following SQL code:

```

    ( select loan_id, (trunc(Date_in) - trunc(Due_date)) * 10 from project1_book_loans
    where (trunc(Due_date) - trunc(Date_in)) < 0);

    commit;

    select * from project1_fines;

    commit;

    select distinct b.title
    from project1_books b, project1_book_copies bc, project1_book_loans bl
    where b.ISBN = bc.ISBN and bc.book_id = bl.book_id
    and bl.Due_date > bl.DATE_IN;

    select a.author_id, a.name, count(b.isbn) as copies
    from project1_authors a join project1_books_authors b on a.author_id=b.author_id
    join project1_books bk on b.isbn=bk.isbn
    group by a.author_id, a.name;
  
```

Below the worksheet is a 'Query Result' window titled 'SQL' which shows the output of the last query:

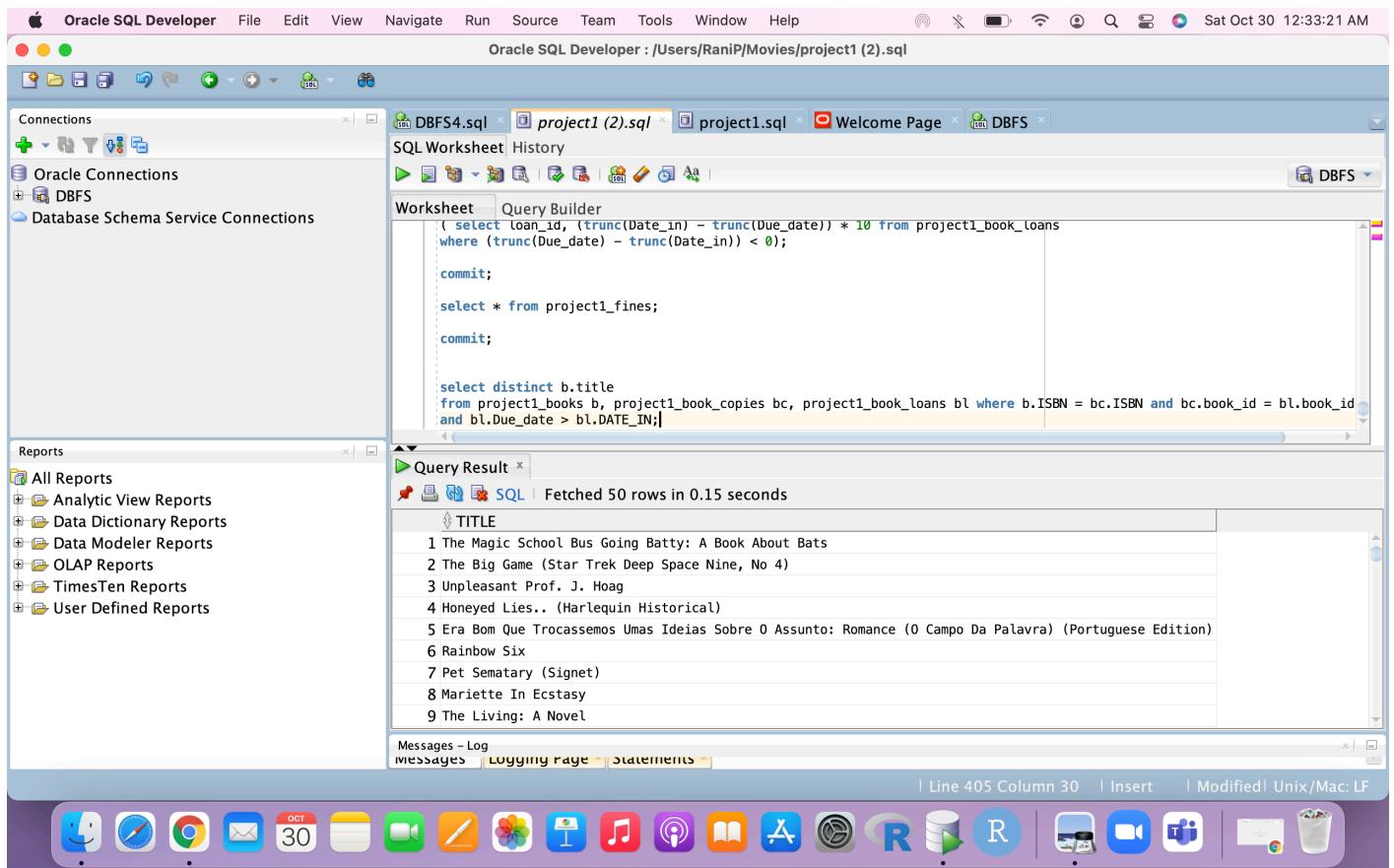
AUTHOR_ID	NAME	COPIES
1	4001 Debbie Macomber	19
2	6196 Jerry B. Jenkins	27
3	8882 Katherine Kurtz	2
4	13257 William Least Heat-Moon	3
5	6554 John Cheever	1
6	2809 Catherine Cookson	1
7	9250 Kristin Hannah	6
8	11040 Pamela Morsi	1
9	11762 Richard Lischer	1
10	14548 Stephanie James	1
11	5672 Isabella MacDonald A	1
12	11095 Patricia A McKillip	1

At the bottom of the interface, there are tabs for 'Messages - Log', 'Logging Page', and 'Statements'. Status indicators at the bottom right show 'Line 409 Column 29', 'Insert', 'Modified', and 'Unix/Mac: LF'.

Report 2: 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.ISBN = bc.ISBN and bc.book_id = bl.book_id
and bl.Due_date > bl.DATE_IN;
  
```



Report 3: Number of each ISBN Publications at the 'Audelia Road' Library Branch

```

select b.isbn,count(bc.book_id)
from project1_books b join project1_book_copies bc on
b.isbn=bc.isbn
join project1_library_branch lb on
bc.branch_id=lb.branch_id where lb.branch_name='Audelia
Road' group by b.isbn;
  
```

Oracle SQL Developer File Edit View Navigate Run Source Team Tools Window Help Fri Oct 29 11:41:30 PM

Oracle SQL Developer : /Users/RaniP/Movies/project1 (2).sql

Connections DBFS4.sql project1 (2).sql project1.sql Welcome Page DBFS

Oracle Connections Database Schema Service Connections

Worksheet Query Builder

```
insert into project1_fines(loan_id, fine_amt)
( select loan_id, (trunc(date_in) - trunc(Due_date)) * 10 from project1_book_loans
where (trunc(Due_date) - trunc(date_in)) < 0);

commit;

select * from project1_fines;

commit;

select b.isbn, count(bc.book_id)
from project1_books b join project1_book_copies bc on b.isbn=bc.isbn
join project1_library_branch lb on bc.branch_id=lb.branch_id where lb.branch_name='Audelia Road' group by b.isbn;
```

Reports Screen Shot 2021-09-28 at 6:34.51 PM

All Reports SQL | Fetched 50 rows in 0.422 seconds

ISBN	COUNT(BC.BOOK_ID)
1 0553278746	1
2 0872280760	1
3 0345430476	1
4 0671018930	1
5 0451452755	1
6 0061030147	1
7 0385337639	1
8 3257216416	1
9 193156146X	1

Messages - Log Logging Page Statements

Line 404 Column 114 | Insert | Modified | Unix/Mac: LF

