



Books Table:


ISBN13	ISBN10	TITLE
9780316973748	0316973742	The Gospel Of Judas: A Novel
9780385235945	0385235941	Prize Stories 1987: The O. Henry Awards
9780446677455	0446677450	Rich Dad, Poor Dad: What The Rich Teach Their Kids About Money--That The Poor And Middle Class Do Not!
9780451166890	0451166892	The Pillars Of The Earth
9780553347593	0553347594	McDonald's: Behind The Arches
9780671621001	0671621009	Creating Wealth: Retire In Ten Years Using Allen's Seven Principles Of Wealth!
9780679763970	067976397X	Corelli's Mandolin: A Novel
9780684822730	0684822733	Love, Miracles, And Animal Healing: A Heartwarming Look At The Spiritual Bond Between Animals And Humans
9780786868711	0786868716	The Five People You Meet In Heaven

book_copy

BOOK_ID	ISBN13	BRANCH_ID	NO_OF_COPIES
051513290X	9780515132908	2	0
051513290X	9780515132908	3	1
051513290X	9780515132908	4	1
051513290X	9780515132908	5	1
0553580388	9780553580389	1	1
0553580388	9780553580389	2	1
0553580388	9780553580389	3	1
0553580388	9780553580389	4	0
0553580388	9780553580389	5	1

gps170000@utdallas.edu

utdallas_034

en

Copyright © 1999, 2018, Oracle. All rights reserved.

Application Express 18.1.0.00.4

Book_authors

ISBN10	AUTHOR_ID
097089726X	29274
0394531094	29275
0394735307	29276
0140043519	29277
0394741919	29278
0618012710	29279
0590603795	29280
0879733438	29281
0590437747	29282

Authors

AUTHOR_NAME	AUTHOR_ID
Jorge Luis Borges	28760
Verity Campbell	28761
Garcia Marquez	28762
S.M. Stirling Anne McCaffrey	28763
Claire Messud	28764
Jerry B. Jenkins	28765
Jeannie. Harmon	28766
Charlie Smith	28767
Barbara Taylor Bradford	28768






Borrowers

CARD_ID	SSN	F_NAME	L_NAME	EMAIL	ADDRESS	CITY	STATE	PHONE	LOAN_ID
ID000264	1	abc	Watkins	jwatkins7b@phoca.cz	9601 Lerdahl Pass	Plano	TX	(214) 814-9170	1500
ID000265	999-21-3141	Andrea	Gonzalez	agonzalez7c@jiathis.com	127 Russell Hill	Dallas	TX	(972) 809-5959	1501
ID000266	999-17-9942	Billy	Little	blittle7d@spiegel.de	46 Autumn Leaf Junction	Dallas	TX	(972) 986-6001	1502
ID000267	999-68-6617	Teresa	Marshall	tmarshall7e@opensource.org	2 Sheridan Trail	Dallas	TX	(469) 884-2394	1503
ID000268	999-20-0144	Benjamin	Lynch	blynch7f@newyorker.com	53 Tennyson Park	Dallas	TX	(469) 877-9851	1504
ID000269	999-35-0300	Nancy	Dixon	ndixon7g@instagram.com	4 Dwight Street	Dallas	TX	(972) 770-3641	1505
ID000270	999-40-6101	Jean	Dunn	jdunn7h@stanford.edu	4 North Street	Dallas	TX	(469) 995-6149	1506
ID000271	999-98-5061	Anthony	Gardner	agardner7i@bizjournals.com	9823 Donald Terrace	Dallas	TX	(972) 432-3180	1507
ID000272	999-84-1386	Benjamin	Morrison	bmorrison7j@parallels.com	21 Oriole Terrace	Dallas	TX	(214) 409-6322	1508

Library_branch

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

Book_loans

EDIT	LOAN_ID	BOOK_ID	CARD_ID	DATE_OUT	DATE_IN	DATE_DUE	BRANCH_ID
	1916	051513290X	ID000041	02/14/2018	02/21/2018	02/28/2018	2
	1893	051513290X	ID000038	01/06/2018	03/01/2018	01/20/2018	2
	2440	051513290X	ID000073	01/13/2018	02/04/2018	01/27/2018	2
	2067	051513290X	ID000171	12/23/2017	01/04/2018	01/06/2018	2
	2322	051513290X	ID000017	02/19/2018	05/07/2018	03/05/2018	2
	2176	051513290X	ID000196	01/08/2018	04/04/2018	01/22/2018	2

Book_fines

LOAN_ID	FINE_AMT	PAID_AMT
2501	0	-
2499	0	-
2500	0	YES
2507	0	-
2508	0	-
1742	16	Yes

Commands used to create each table

Books:

```
create table book as select ISBN13,ISBN10,TITLE from book_non_normalized;
```

book_copy

```
CREATE TABLE BOOK_COPY AS SELECT B.ISBN10 AS BOOK_ID ,  
B.ISBN13,A.BRANCH_ID,A.NO_OF_COPIES from book_non_normalized B join  
book_copies_non_normalized A on B.ISBN10=A.BOOK_ID;
```

Library_branch

```
select * from lib_non_normal;
```

```
alter table lib_non_normal add primary key(BRANCH_ID);
```

borrowers:

```
create table borrowers as select ID0000ID AS CARD_ID,SSN,FIRST_NAME AS  
F_NAME,LAST_NAME AS L_NAME,EMAIL,ADDRESS,CITY,STATE,PHONE from  
BORROWERS_NON_NORMALIZED;
```

```
alter table borrowers add (loan_id int);
```

```
create sequence loan_id
```

```
start with 1000 increment by 1
```

```
maxvalue 999999999999
```

```
minvalue 1
```

```
nocycle;
```

```
update book_loans set loan_id=loan_id.NEXTVAL;
```

```
book_loans
```

```
create table extreme_temp as (select loan_id from borrowers);
```

```
alter table extreme_temp add(date_out date);
```

```
book_loans.date_in
```

```
begin
```

```
for i in (select * from extreme_temp) loop
```

```
update extreme_temp set date_out= (sysdate-365+trunc(dbms_random.value(1,100))) ;
```

```
end loop;
```

```
end;
```

```
book_loans.date_in
```

```
begin
```

```
for i in (select loan_id as loan_id from extreme_temp) loop
```

```
update extreme_temp set date_in= date_out + dbms_random.value(1,100) where  
loan_id=i.loan_id ;
```

```
end loop;
```

```
end;
```

```
book_loans.date_due
```

```
begin
```

```
for i in (select loan_id as loan_id from extreme_temp) loop
```

```
update extreme_temp set date_due= date_out+14 where loan_id=i.loan_id ;
```

```
end loop;
```

```
end;
```

Part 3: Normalization

The below code is used to normalized the author in books original table so that all of them fall into different rows

```
insert into temp_author_normal (ISBN10,author_name)
```

```
select ISBN,part_1 from (
```

```
SELECT b.*,B.ISBN10 AS ISBN ,REGEXP_SUBSTR (B.author, '^[,]+', 1, 1) AS part_1
```

```
FROM BOOK_non_normalized b) where part_1 is not null;
```

```
insert into temp_author_normal (ISBN10,author_name)
```

```
select ISBN,part_1 from (
```

```
SELECT b.*,B.ISBN10 AS ISBN ,REGEXP_SUBSTR (B.author, '^[,]+', 1, 2) AS part_1
```

```
FROM BOOK_non_normalized b) where part_1 is not null;
```

```
insert into temp_author_normal (ISBN10,author_name)
```

```
select ISBN,part_1 from (
```

```
SELECT b.*,B.ISBN10 AS ISBN ,REGEXP_SUBSTR (B.author, '^[,]+', 1, 3) AS part_1
```

```
FROM BOOK_non_normalized b) where part_1 is not null;
```

```

insert into temp_author_normal (ISBN10,author_name)
select ISBN,part_1 from (
SELECT b.*,B.ISBN10 AS ISBN ,REGEXP_SUBSTR (B.author, '^[,]+' , 1, 4) AS part_1
FROM BOOK_non_normalized b) where part_1 is not null;

```

```

insert into temp_author_normal (ISBN10,author_name)
select ISBN,part_1 from (
SELECT b.*,B.ISBN10 AS ISBN ,REGEXP_SUBSTR (B.author, '^[,]+' , 1, 5) AS part_1
FROM BOOK_non_normalized b) where part_1 is not null;

```

Part 4: Populating tables book_loans and book_fines

Book_loans

```

create table book_loans as select * from(select loan_id,card_no,book_id,
date_out,date_in,date_due
from
(select book_id from book_copy where rownum<100) ,
(select ID0000ID AS card_no from borrowers where rownum <200),
(select loan_id,date_in,date_out,date_due from extreme_temp )
ORDER BY DBMS_RANDOM.VALUE) where
rownum<500;

```

book_fines

```

create table loan_id_fines (loan_id int,date_in date,date_out date,date_due date);

```

```
insert into loan_id_fines(SELECT * FROM (select loan_id,date_in,date_out,date_due from
(select loan_id,date_in,date_out,date_due from BOOK_LOANS ),(select card_id from
borrowers where rownum<20)
```

```
order by DBMS_RANDOM.VALUE) WHERE ROWNUM<50);
```

```
create table fine_amt as (select (date_due -date_in)*0.25 as fine_amt from loan_id_fines);
```

```
alter table book_fines drop column paid_amt;
```

```
alter table book_fines drop primary key;
```

```
insert into book_fines( select l.loan_id,f.fine_amt from loan_id_fines l cross join fine_amt f
where rownum<49);
```

Part 5: Triggers

1. The below trigger will check if there is any fine due or if the person has checked out more than three books and will insert the table accordingly.

```
create or replace TRIGGER book_loans_insert_with_branch
```

```
BEFORE INSERT
```

```
ON "BOOK_LOANS"
```

```
FOR EACH ROW
```

```
declare
```

```
card_no_count NUMBER(38);
```

```
loan_id int;
```

```
over_due_fine number(38);
```

```
BEGIN
```

```
SELECT COUNT(*) INTO CARD_NO_COUNT FROM BOOK_LOANS WHERE CARD_ID
=:NEW.CARD_ID;
```

```

select max(loan_id)+1 into loan_id from book_loans;

select sum(book_fines.fine_amt) INTO over_due_fine from book_loans join book_fines on
book_loans.loan_id=book_fines.loan_id where book_loans.card_ID=:NEW.CARD_ID;

IF card_no_COUNT<3 THEN

:NEW.LOAN_ID:=loan_id;

:NEW.DATE_OUT:=sysdate;

:NEW.DATE_IN:=NULL;

:NEW.DATE_DUE:=SYSDATE+14;

insert into BOOK_FINES(loan_id,fine_amt,paid_amt) values(loan_id,0,"");

else if card_no_COUNT>=3 then

:NEW.DATE_OUT:=NULL;

:NEW.DATE_IN:=NULL;

:NEW.DATE_DUE:=NULL;

:NEW.LOAN_ID:=null;

:NEW.CARD_ID:=NULL;

:NEW.BOOK_ID:=NULL;

delete from book_loans where LOAN_ID IS null;

RAISE_APPLICATION_ERROR(-20001,' more than three books withdrawn ');

END IF;

END IF;

IF over_due_fine>0 then

:NEW.DATE_OUT:=NULL;

:NEW.DATE_IN:=NULL;

:NEW.DATE_DUE:=NULL;

:NEW.LOAN_ID:=null;

:NEW.CARD_ID:=NULL;

```



```

:NEW.BOOK_ID:=NULL;

delete from book_loans where LOAN_ID IS null;

RAISE_APPLICATION_ERROR(-20001,'fine due ');

end if;


END;

```

2. Trigger on borrower table when insert occurs .

```

create or replace TRIGGER borrowers_on_insert
BEFORE INSERT
ON borrowers
FOR EACH ROW
declare
LOAN_ID INT;
CARD_ID VARCHAR(100);
begin
SELECT MAX(LOAN_ID)+1 INTO LOAN_ID FROM BORROWERS;
SELECT CONCAT('ID00',MAX(LOAN_ID)+1) INTO CARD_ID FROM BORROWERS;
:NEW.LOAN_ID:=LOAN_ID;
:NEW.CARD_ID:=CARD_ID;

END;

```

3. Trigger to check if book is available in that branch and if checked out to reduce the quantity by 1

```

create or replace TRIGGER book_loans_insert_branch
BEFORE INSERT
ON "BOOK_LOANS"
FOR EACH ROW
declare
no_of_copies int;

```

```

begin

select book_copy.no_of_copies into no_of_copies from book_copy where
book_copy.book_id=:NEW.BOOK_ID AND book_copy.BRANCH_ID=:NEW.BRANCH_ID;

if no_of_copies>0 THEN

update book_copy set no_of_copies= no_of_copies-1 WHERE
book_copy.book_id=:NEW.BOOK_ID AND book_copy.BRANCH_ID=:NEW.BRANCH_ID;

ELSE

:NEW.DATE_OUT:=NULL;

:NEW.DATE_IN:=NULL;

:NEW.DATE_DUE:=NULL;

:NEW.LOAN_ID:=null;

:NEW.CARD_ID:=NULL;

:NEW.BOOK_ID:=NULL;

delete from book_loans where LOAN_ID IS null;

RAISE_APPLICATION_ERROR(-20001,' BOOK NOT AVAILABLE IN THAT BRANCH ');

END IF;

END;

```

4. Trigger to update the fine cost

```

CREATE OR REPLACE TRIGGER book_fines_before_insert
after insert
ON fines_check
for each row
declare
loan_id int;
date_diff int;
begin
update book_loans set DATE_IN=sysdate where loan_id=:NEW.LOAN_ID;

```

```
select (trunc(date_in-date_out)*0.25) into date_diff from book_loans where  
loan_id=:NEW.LOAN_ID;
```

```
update FINES_CHECK set fine_amt=date_diff;
```

```
END;
```

```
update book_loans_new set loan_id=3785 where loan_id=3785;
```

```
select * from book_loans where loan_id=3785;
```

note: this update statement will make the the date_in for that record as sysdate i.e todays date
and will also calculate the fine amt and update in fines table