Project 1

Members

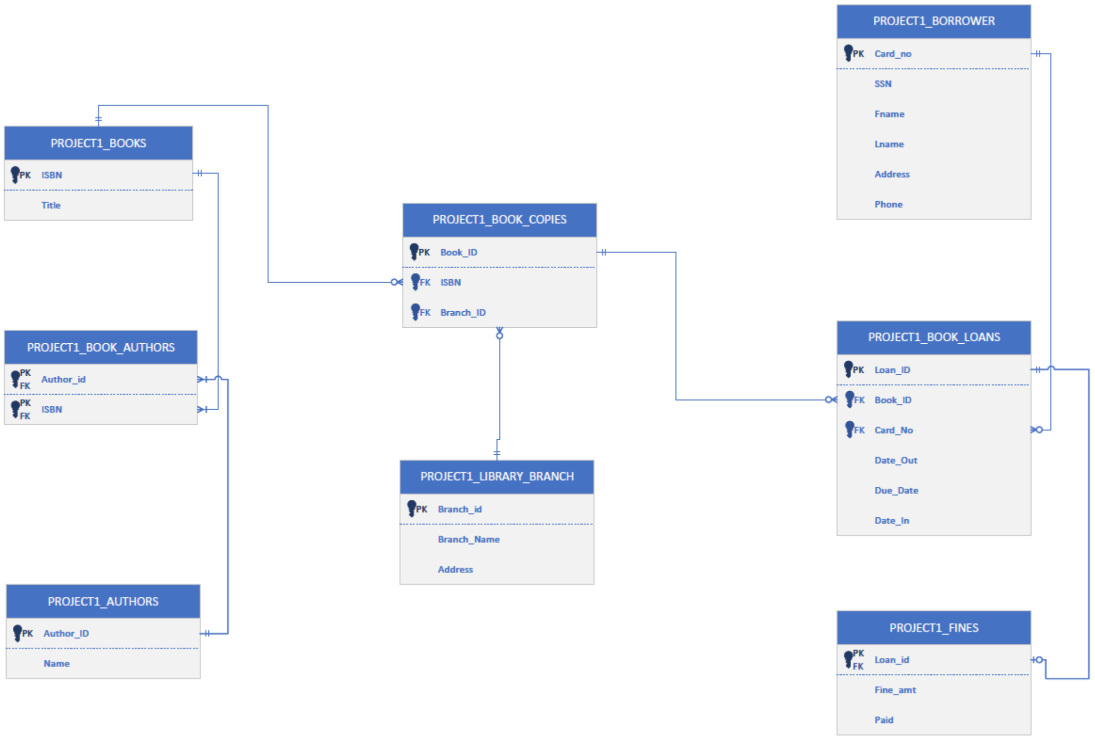
Karan Suhas Chiddarwar | Net ID: KXC190064

Pranshul Tomar | Net ID: PXT200004

Taranjeet Singh Gill | Net ID: TXG210007

Design and DB Architecture

1. ***ER diagram***



2. ***Database objects create statements: create table, foreign keys, indexes, etc***

   - Book

```
CREATE TABLE project1_books (
isbn VARCHAR2(26 BYTE),
title VARCHAR2(1024 BYTE),
PRIMARY KEY (isbn));
```

   - Book_Authors

```
CREATE TABLE project1_book_authors (
author_id NUMBER(38,0),
isbn VARCHAR2(26 BYTE));

--Primary Key Addition
```

```
CREATE INDEX book_authors_idx ON project1_book_authors(author_id,isbn);
ALTER TABLE project1_book_authors ADD CONSTRAINT pk_authorID_isbn
PRIMARY KEY (author_id, isbn);

--Foreign Key Addition
ALTER TABLE project1_book_authors
ADD CONSTRAINT AuthorID_FK
FOREIGN KEY (author_id) REFERENCES project1_authors(author_id);

ALTER TABLE project1_book_authors
ADD CONSTRAINT ISBN_FK
FOREIGN KEY (isbn) REFERENCES project1_books(isbn);
```

- Authors

```
CREATE TABLE project1_authors (
author_id NUMBER GENERATED ALWAYS as IDENTITY(START with 100001
INCREMENT by 1),
name VARCHAR2(1024 BYTE),
PRIMARY KEY (author_id));
```

- Library_Branch

```
CREATE TABLE project1_library_branch (
branch_id NUMBER(38,0),
branch_name VARCHAR2(26 BYTE),
address VARCHAR2(128 BYTE),
PRIMARY KEY (branch_id));
```

- Book_Copies

```
CREATE TABLE project1_book_copies (
book_id NUMBER GENERATED ALWAYS as IDENTITY(START with 200001 INCREMENT
by 1),
isbn VARCHAR2(26 BYTE),
branch_id NUMBER(38,0),
PRIMARY KEY (book_id));

--Foreign Key Addition
ALTER TABLE project1_book_copies
ADD CONSTRAINT ISBN_BC_FK
FOREIGN KEY (isbn) REFERENCES project1_books(isbn);

ALTER TABLE project1_book_copies
ADD CONSTRAINT BranchID_FK
FOREIGN KEY (branch_id) REFERENCES project1_library_branch(branch_id);
```

- Borrower

```sql
CREATE TABLE project1_borrower (
card_no VARCHAR2(26 BYTE),
ssn VARCHAR2(26 BYTE),
fname VARCHAR2(26 BYTE),
lname VARCHAR2(26 BYTE),
address VARCHAR2(1024 BYTE),
phone VARCHAR2(26 BYTE),
PRIMARY KEY (card_no));
```

- Book_Loans

```sql
CREATE TABLE project1_book_loans (
loan_id NUMBER GENERATED ALWAYS as IDENTITY(START with 300001 INCREMENT by 1),
book_id NUMBER(38,0),
card_no VARCHAR2(26 BYTE),
date_out DATE,
due_date DATE,
date_in DATE,
PRIMARY KEY (loan_id));

--Foreign Key Addition
ALTER TABLE project1_book_loans
ADD CONSTRAINT BookID_FK
FOREIGN KEY (book_id) REFERENCES project1_book_copies(book_id);

ALTER TABLE project1_book_loans
ADD CONSTRAINT CardNO_FK
FOREIGN KEY (card_no) REFERENCES project1_borrower(card_no);
```

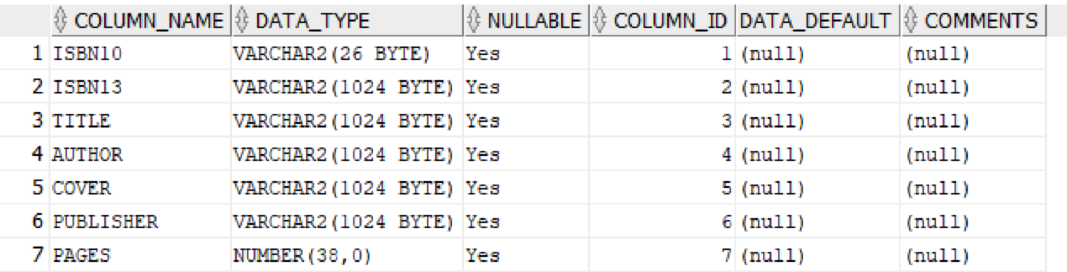
- Fines

```sql
CREATE TABLE project1_fines (
loan_id NUMBER,
fine_amt NUMBER(38,1),
paid VARCHAR2(26 BYTE));

--Primary Key Addition
ALTER TABLE project1_fines
ADD CONSTRAINT LoanID_PK PRIMARY KEY (loan_id);

--Foreign Key Addition
ALTER TABLE project1_fines
ADD CONSTRAINT LoanID_FK
FOREIGN KEY (loan_id) REFERENCES project1_book_loans(loan_id);
```

Data Load, Normalization, data generation

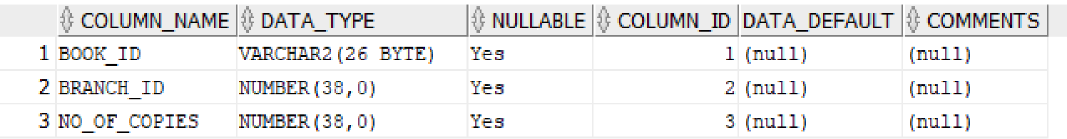
1. ***Loading Input Files***

   - Books_Load



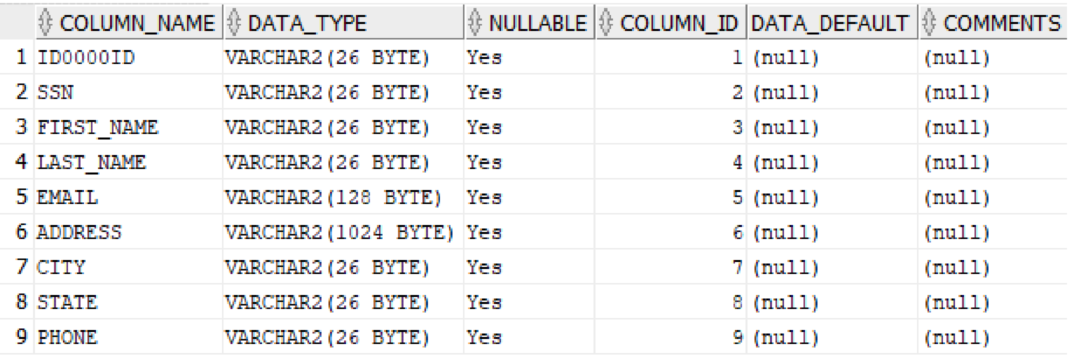
	COLUMN_NAME	DATA_TYPE	NULLABLE	COLUMN_ID	DATA_DEFAULT	COMMENTS
1	ISBN10	VARCHAR2(26 BYTE)	Yes	1	(null)	(null)
2	ISBN13	VARCHAR2(1024 BYTE)	Yes	2	(null)	(null)
3	TITLE	VARCHAR2(1024 BYTE)	Yes	3	(null)	(null)
4	AUTHOR	VARCHAR2(1024 BYTE)	Yes	4	(null)	(null)
5	COVER	VARCHAR2(1024 BYTE)	Yes	5	(null)	(null)
6	PUBLISHER	VARCHAR2(1024 BYTE)	Yes	6	(null)	(null)
7	PAGES	NUMBER(38,0)	Yes	7	(null)	(null)

   - Book_Copies_Load



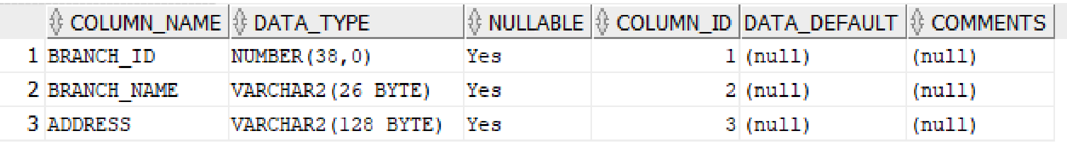
	COLUMN_NAME	DATA_TYPE	NULLABLE	COLUMN_ID	DATA_DEFAULT	COMMENTS
1	BOOK_ID	VARCHAR2(26 BYTE)	Yes	1	(null)	(null)
2	BRANCH_ID	NUMBER(38,0)	Yes	2	(null)	(null)
3	NO_OF_COPIES	NUMBER(38,0)	Yes	3	(null)	(null)

   - Borrowers_Load



	COLUMN_NAME	DATA_TYPE	NULLABLE	COLUMN_ID	DATA_DEFAULT	COMMENTS
1	ID0000ID	VARCHAR2(26 BYTE)	Yes	1	(null)	(null)
2	SSN	VARCHAR2(26 BYTE)	Yes	2	(null)	(null)
3	FIRST_NAME	VARCHAR2(26 BYTE)	Yes	3	(null)	(null)
4	LAST_NAME	VARCHAR2(26 BYTE)	Yes	4	(null)	(null)
5	EMAIL	VARCHAR2(128 BYTE)	Yes	5	(null)	(null)
6	ADDRESS	VARCHAR2(1024 BYTE)	Yes	6	(null)	(null)
7	CITY	VARCHAR2(26 BYTE)	Yes	7	(null)	(null)
8	STATE	VARCHAR2(26 BYTE)	Yes	8	(null)	(null)
9	PHONE	VARCHAR2(26 BYTE)	Yes	9	(null)	(null)

   - Library_Branch_Load



	COLUMN_NAME	DATA_TYPE	NULLABLE	COLUMN_ID	DATA_DEFAULT	COMMENTS
1	BRANCH_ID	NUMBER(38,0)	Yes	1	(null)	(null)
2	BRANCH_NAME	VARCHAR2(26 BYTE)	Yes	2	(null)	(null)
3	ADDRESS	VARCHAR2(128 BYTE)	Yes	3	(null)	(null)

2. ***SQL scripts to populate data in target application tables from data provided in Load Files***

   - Book

```
INSERT INTO project1_books (isbn, title)
SELECT TRIM(isbn10), TRIM(title) FROM project1_books_load;

SELECT COUNT(*) FROM project1_books; -- 25,000
```

   - Authors

```
--Get Distinct Authors from books_load table
--Creating a Dummy Table to temporarily hold values
--book_authors_dummy table contains unique list of authors along with
their book(isbn)
CREATE TABLE book_authors_dummy (
author VARCHAR2(1024 BYTE),
isbn VARCHAR2(26 BYTE));

--PL/SQL to fetch comma seperated Authors and store that in
book_authors_dummy
BEGIN
FOR temp IN (SELECT author, isbn10 FROM project1_books_load)
LOOP
    IF (regexp_count(temp.author, ',') + 1) = 1 THEN
        INSERT INTO book_authors_dummy (author, isbn) VALUES
(temp.author, temp.isbn10);
    ELSIF (regexp_count(temp.author, ',') + 1) = 2 THEN
        INSERT INTO book_authors_dummy (author, isbn) VALUES
(regexp_substr(temp.author, '[^,]+', 1, 1), temp.isbn10);
        INSERT INTO book_authors_dummy (author, isbn) VALUES
(regexp_substr(temp.author, '[^,]+', 1, 2), temp.isbn10);
    ELSIF (regexp_count(temp.author, ',') + 1) = 3 THEN
        INSERT INTO book_authors_dummy (author, isbn) VALUES
(regexp_substr(temp.author, '[^,]+', 1, 1), temp.isbn10);
        INSERT INTO book_authors_dummy (author, isbn) VALUES
(regexp_substr(temp.author, '[^,]+', 1, 2), temp.isbn10);
        INSERT INTO book_authors_dummy (author, isbn) VALUES
(regexp_substr(temp.author, '[^,]+', 1, 3), temp.isbn10);
    ELSIF (regexp_count(temp.author, ',') + 1) = 4 THEN
        INSERT INTO book_authors_dummy (author, isbn) VALUES
(regexp_substr(temp.author, '[^,]+', 1, 1), temp.isbn10);
        INSERT INTO book_authors_dummy (author, isbn) VALUES
(regexp_substr(temp.author, '[^,]+', 1, 2), temp.isbn10);
        INSERT INTO book_authors_dummy (author, isbn) VALUES
(regexp_substr(temp.author, '[^,]+', 1, 3), temp.isbn10);
        INSERT INTO book_authors_dummy (author, isbn) VALUES
(regexp_substr(temp.author, '[^,]+', 1, 4), temp.isbn10);
    ELSIF (regexp_count(temp.author, ',') + 1) = 5 THEN
        INSERT INTO book_authors_dummy (author, isbn) VALUES
(regexp_substr(temp.author, '[^,]+', 1, 1), temp.isbn10);
        INSERT INTO book_authors_dummy (author, isbn) VALUES
(regexp_substr(temp.author, '[^,]+', 1, 2), temp.isbn10);
        INSERT INTO book_authors_dummy (author, isbn) VALUES
(regexp_substr(temp.author, '[^,]+', 1, 3), temp.isbn10);
        INSERT INTO book_authors_dummy (author, isbn) VALUES
(regexp_substr(temp.author, '[^,]+', 1, 4), temp.isbn10);
        INSERT INTO book_authors_dummy (author, isbn) VALUES
(regexp_substr(temp.author, '[^,]+', 1, 5), temp.isbn10);
    END IF;
END LOOP;
END;
/


--Insert distinct authors from book_authors_dummy to project1_authors
table

--author_id is defined as "author_id NUMBER GENERATED ALWAYS as
IDENTITY(START with 100001 INCREMENT by 1" hence a unique value will be
assigned automatically when a record is created

INSERT INTO project1_authors (name)
SELECT DISTINCT author FROM book_authors_dummy;
SELECT COUNT(*) FROM project1_authors; -- 15649
```

- Book_Author

```
--INSERTING values in book_authors table by joining authors table (which
contains author_id) with book_authors_dummy (which contains author name
and isbn)

INSERT INTO project1_book_authors (author_id, isbn)
SELECT DISTINCT temp.author_id, temp.isbn FROM(
SELECT ba_dummy.author AS author_name, ba_dummy.isbn AS isbn,
a.author_id AS author_id FROM book_authors_dummy ba_dummy
INNER JOIN project1_authors a ON TRIM(a.name) = TRIM(ba_dummy.author))
temp;

SELECT COUNT(*) FROM project1_book_authors; --30240
```

- Library_Branch

```
INSERT INTO project1_library_branch
SELECT * FROM project1_library_branch_load;

SELECT COUNT(*) FROM project1_library_branch; --5
```

- Book_Copies

```
--PL SQL to insert data into Book_Copies table from Book_Copies_Load

--book_id is defined as "book_id NUMBER GENERATED ALWAYS as
IDENTITY(START with 200001 INCREMENT by 1" hence a unique value will be
assigned automatically when a record is created

BEGIN
FOR temp IN (SELECT book_id, branch_id, no_of_copies FROM
project1_book_copies_load WHERE no_of_copies <> 0 ORDER BY book_id,
branch_id)
LOOP
FOR i IN 1..temp.no_of_copies
LOOP
INSERT INTO project1_book_copies (isbn,branch_id) VALUES (temp.book_id,
temp.branch_id);
END LOOP;
END LOOP;
END;
/

SELECT COUNT(*) FROM project1_book_copies; --94338
```

- Borrower

```
INSERT INTO project1_borrower(card_no, ssn, fname, lname, address,
phone)
SELECT id0000id, ssn, first_name, last_name, address, phone FROM
project1_borrowers_load;

SELECT COUNT(*) FROM project1_borrower; --1000
```

3. ***SQL to generate 400 books check-outs for exactly 200 different borrowers and exactly 100 different books & Exactly 50 fines records for 50 different borrowers***
    - Book_Loans

```
--Insert Recors In Book_loans

DECLARE

startDate  DATE := TO_DATE('01-JAN-2012');
i NUMBER := 0;
row_number NUMBER := 1;

BEGIN

FOR temp IN (SELECT c_no AS borrower, b_id AS book FROM(
SELECT b.*, bc.*, DENSE_RANK() OVER(order by c_no) as borrower_rank,
ROW_NUMBER() OVER(partition by c_no order by c_no,b_id) as
borrower_book_rank FROM(
SELECT * FROM(
SELECT *
FROM (
SELECT bc.book_id AS b_id
FROM project1_book_copies bc
ORDER BY DBMS_RANDOM.RANDOM)
WHERE ROWNUM<=100)) bc,
(SELECT * FROM(
SELECT *
FROM (
SELECT b.card_no AS c_no
FROM project1_borrower b
ORDER BY DBMS_RANDOM.RANDOM)
WHERE ROWNUM<=200)) b)
WHERE borrower_rank = borrower_book_rank OR borrower_rank =
borrower_book_rank - 1 OR (borrower_rank = 100 AND borrower_book_rank =
1)
OR borrower_rank = borrower_book_rank + 100 OR borrower_rank =
borrower_book_rank + 101 OR (borrower_rank = 101 AND borrower_book_rank
= 2))

LOOP

IF (row_number > 0 AND row_number <= 300) THEN
INSERT INTO project1_book_loans
(book_id,card_no,date_out,due_date,date_in) VALUES (temp.book,
temp.borrower,(startDate+i),(startDate+i+2),(startDate+i+2));
```

```
i := i+3;
row_number := row_number+1;

ELSIF (row_number > 300 AND row_number <= 310 AND MOD(row_number, 2) <>
0) THEN
INSERT INTO project1_book_loans
(book_id,card_no,date_out,due_date,date_in) VALUES (temp.book,
temp.borrower,(startDate+i),(startDate+i+2),(startDate+i+3));
i := i+3;
row_number := row_number+1;

ELSIF (row_number > 310 AND row_number <= 320 AND MOD(row_number, 2) <>
0) THEN
INSERT INTO project1_book_loans
(book_id,card_no,date_out,due_date,date_in) VALUES (temp.book,
temp.borrower,(startDate+i),(startDate+i+2),(startDate+i+4));
i := i+4;
row_number := row_number+1;

ELSIF (row_number > 320 AND row_number <= 330 AND MOD(row_number, 2) <>
0) THEN
INSERT INTO project1_book_loans
(book_id,card_no,date_out,due_date,date_in) VALUES (temp.book,
temp.borrower,(startDate+i),(startDate+i+2),(startDate+i+5));
i := i+4;
row_number := row_number+1;

ELSIF (row_number > 330 AND row_number <= 340 AND MOD(row_number, 2) <>
0) THEN
INSERT INTO project1_book_loans
(book_id,card_no,date_out,due_date,date_in) VALUES (temp.book,
temp.borrower,(startDate+i),(startDate+i+2),(startDate+i+6));
i := i+5;
row_number := row_number+1;

ELSIF (row_number > 340 AND row_number <= 350 AND MOD(row_number, 2) <>
0) THEN
INSERT INTO project1_book_loans
(book_id,card_no,date_out,due_date,date_in) VALUES (temp.book,
temp.borrower,(startDate+i),(startDate+i+2),(startDate+i+7));
i := i+5;
row_number := row_number+1;

ELSIF (row_number > 350 AND row_number <= 360 AND MOD(row_number, 2) <>
0) THEN
INSERT INTO project1_book_loans
(book_id,card_no,date_out,due_date,date_in) VALUES (temp.book,
temp.borrower,(startDate+i),(startDate+i+2),(startDate+i+8));
i := i+6;
row_number := row_number+1;

ELSIF (row_number > 360 AND row_number <= 370 AND MOD(row_number, 2) <>
0) THEN
INSERT INTO project1_book_loans
(book_id,card_no,date_out,due_date,date_in) VALUES (temp.book,
temp.borrower,(startDate+i),(startDate+i+2),(startDate+i+9));
i := i+6;
row_number := row_number+1;
```

```
ELSIF (row_number > 370 AND row_number <= 380 AND MOD(row_number, 2) <>
0) THEN
INSERT INTO project1_book_loans
(book_id,card_no,date_out,due_date,date_in) VALUES (temp.book,
temp.borrower,(startDate+i),(startDate+i+2),(startDate+i+10));
i := i+7;
row_number := row_number+1;

ELSIF (row_number > 380 AND row_number <= 390 AND MOD(row_number, 2) <>
0) THEN
INSERT INTO project1_book_loans
(book_id,card_no,date_out,due_date,date_in) VALUES (temp.book,
temp.borrower,(startDate+i),(startDate+i+2),(startDate+i+11));
i := i+7;
row_number := row_number+1;

ELSIF (row_number > 390 AND row_number <= 400 AND MOD(row_number, 2) <>
0) THEN
INSERT INTO project1_book_loans
(book_id,card_no,date_out,due_date,date_in) VALUES (temp.book,
temp.borrower,(startDate+i),(startDate+i+2),(startDate+i+12));
i := i+8;
row_number := row_number+1;

ELSE
INSERT INTO project1_book_loans
(book_id,card_no,date_out,due_date,date_in) VALUES (temp.book,
temp.borrower,(startDate+i),(startDate+i+2),(startDate+i+2));
i := i+8;
row_number := row_number+1;

END IF;
END LOOP;
END;
/


--Validating exactly 200 Unique borrowers in Book Loans Table
SELECT COUNT(DISTINCT card_no) FROM project1_book_loans; --200

--Validating exactly 100 Unique books in Book Loans Table
SELECT COUNT(DISTINCT book_id) FROM project1_book_loans; --100

--Validating exactly 400 records in Book Loans Table
SELECT COUNT(*) FROM project1_book_loans; --400
```

- Fines

```
--Insert Records in Fines Table

BEGIN
```

```
FOR temp IN (SELECT loan_id, book_id, card_no, date_out, due_date,
date_in, TO_NUMBER(date_in - due_date) AS fine_amount, MOD(ROWNUM, 3) AS
is_fine_applicable  FROM project1_book_loans WHERE ((date_in - due_date)
>= 1))
    LOOP
        IF temp.is_fine_applicable = 0 THEN
        INSERT INTO project1_fines (loan_id, fine_amt, paid) VALUES
(temp.loan_id, temp.fine_amount * 15, 'NO');
        ELSE
        INSERT INTO project1_fines (loan_id, fine_amt, paid) VALUES
(temp.loan_id, temp.fine_amount * 15, 'YES');
        END IF;
    END LOOP;

END;
/

--Validating exactly 50 records in Fines Table
SELECT COUNT(*) FROM project1_fines; --50

--Validating exactly 50 unique card_no in Fines Table
SELECT COUNT(DISTINCT temp.card_no) FROM(
SELECT bl.card_no AS card_no, f.loan_id FROM project1_book_loans bl
INNER JOIN project1_fines f ON bl.loan_id = f.loan_id) temp; --50
```

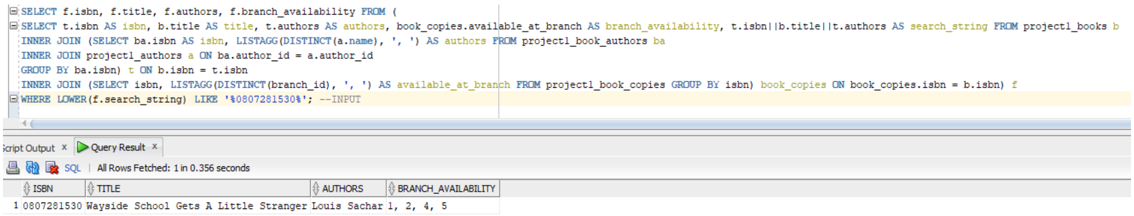
Book Search and Availability

Add either value of ISBN, Ttile, Author Name, substring of Title or substring of Author Name in INPUT and Branch availability will be displayed in 'BRANCH_AVAILABILITY'.

This is a single search functionality to locate a book given any combination of ISBN, title, and/or Author(s) and their substring

- SQL Search when input is ISBN

```
SELECT f.isbn, f.title, f.authors, f.branch_availability FROM (
SELECT t.isbn AS isbn, b.title AS title, t.authors AS authors,
book_copies.available_at_branch AS branch_availability,
t.isbn||b.title||t.authors AS search_string FROM project1_books b
INNER JOIN (SELECT ba.isbn AS isbn, LISTAGG(DISTINCT(a.name), ', ') AS
authors FROM project1_book_authors ba
INNER JOIN project1_authors a ON ba.author_id = a.author_id
GROUP BY ba.isbn) t ON b.isbn = t.isbn
INNER JOIN (SELECT isbn, LISTAGG(DISTINCT(branch_id), ', ') AS
available_at_branch FROM project1_book_copies GROUP BY isbn) book_copies ON
book_copies.isbn = b.isbn) f
WHERE LOWER(f.search_string) LIKE '%0807281530%'; --INPUT
```



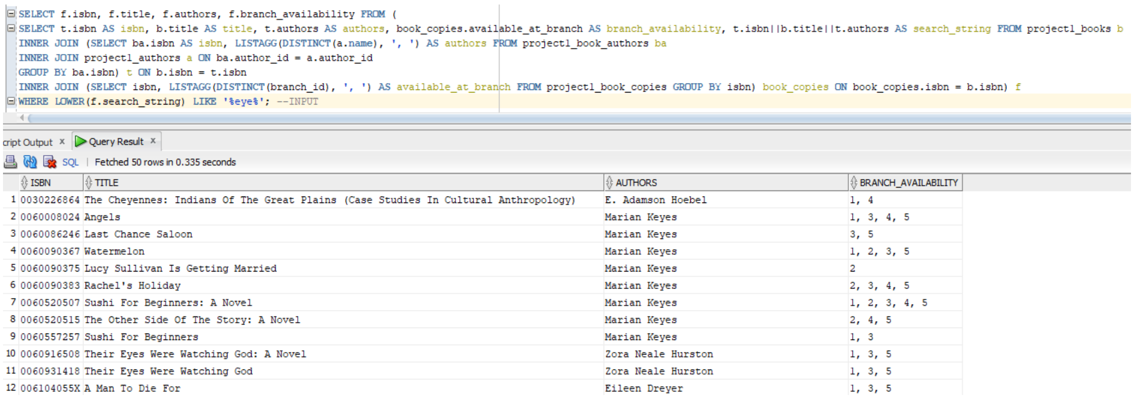
```
SELECT f.isbn, f.title, f.authors, f.branch_availability FROM (
SELECT t.isbn AS isbn, b.title AS title, t.authors AS authors, book_copies.available_at_branch AS branch_availability, t.isbn||b.title||t.authors AS search_string FROM project1_books b
INNER JOIN (SELECT ba.isbn AS isbn, LISTAGG(DISTINCT(a.name), ', ') AS authors FROM project1_book_authors ba
INNER JOIN project1_authors a ON ba.author_id = a.author_id
GROUP BY ba.isbn) t ON b.isbn = t.isbn
INNER JOIN (SELECT isbn, LISTAGG(DISTINCT(branch_id), ', ') AS available_at_branch FROM project1_book_copies GROUP BY isbn) book_copies ON book_copies.isbn = b.isbn) f
WHERE LOWER(f.search_string) LIKE '%0807281530%'; --INPUT
```

All Rows Fetched: 1 in 0.356 seconds

	ISBN	TITLE	AUTHORS	BRANCH_AVAILABILITY
1	0807281530	Wayside School Gets A Little Stranger	Louis Sachar	1, 2, 4, 5

- SQL Search when input is Title or substring of Title

```sql
SELECT f.isbn, f.title, f.authors, f.branch_availability FROM (
SELECT t.isbn AS isbn, b.title AS title, t.authors AS authors,
book_copies.available_at_branch AS branch_availability,
t.isbn||b.title||t.authors AS search_string FROM project1_books b
INNER JOIN (SELECT ba.isbn AS isbn, LISTAGG(DISTINCT(a.name), ', ') AS
authors FROM project1_book_authors ba
INNER JOIN project1_authors a ON ba.author_id = a.author_id
GROUP BY ba.isbn) t ON b.isbn = t.isbn
INNER JOIN (SELECT isbn, LISTAGG(DISTINCT(branch_id), ', ') AS
available_at_branch FROM project1_book_copies GROUP BY isbn) book_copies ON
book_copies.isbn = b.isbn) f
WHERE LOWER(f.search_string) LIKE '%eye%'; --INPUT
```



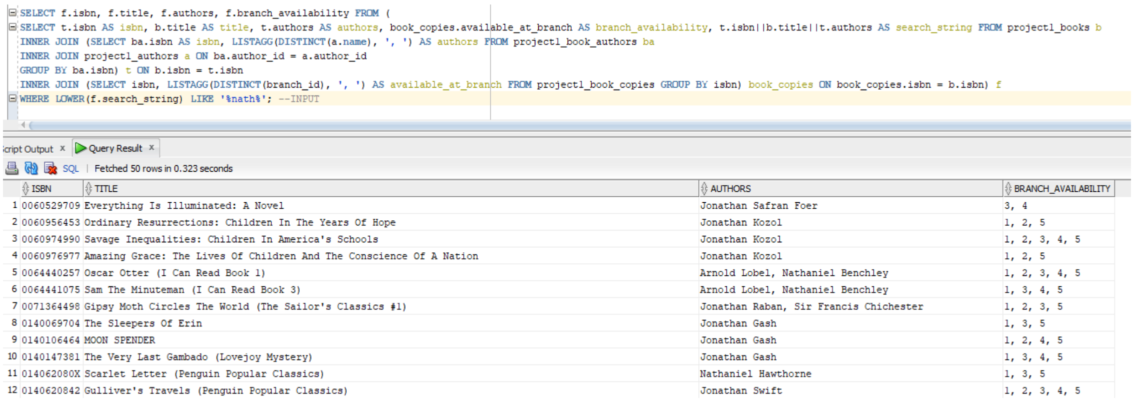
```sql
SELECT f.isbn, f.title, f.authors, f.branch_availability FROM (
SELECT t.isbn AS isbn, b.title AS title, t.authors AS authors, book_copies.available_at_branch AS branch_availability, t.isbn||b.title||t.authors AS search_string FROM project1_books b
INNER JOIN (SELECT ba.isbn AS isbn, LISTAGG(DISTINCT(a.name), ', ') AS authors FROM project1_book_authors ba
INNER JOIN project1_authors a ON ba.author_id = a.author_id
GROUP BY ba.isbn) t ON b.isbn = t.isbn
INNER JOIN (SELECT isbn, LISTAGG(DISTINCT(branch_id), ', ') AS available_at_branch FROM project1_book_copies GROUP BY isbn) book_copies ON book_copies.isbn = b.isbn) f
WHERE LOWER(f.search_string) LIKE '%eye%'; --INPUT
```

Fetched 50 rows in 0.335 seconds

	ISBN	TITLE	AUTHORS	BRANCH_AVAILABILITY
1	0030226864	The Cheyennes: Indians Of The Great Plains (Case Studies In Cultural Anthropology)	E. Adamson Hoebel	1, 4
2	0060008024	Angels	Marian Keyes	1, 3, 4, 5
3	0060086246	Last Chance Saloon	Marian Keyes	3, 5
4	0060090367	Watermelon	Marian Keyes	1, 2, 3, 5
5	0060090375	Lucy Sullivan Is Getting Married	Marian Keyes	2
6	0060090383	Rachel's Holiday	Marian Keyes	2, 3, 4, 5
7	0060520507	Sushi For Beginners: A Novel	Marian Keyes	1, 2, 3, 4, 5
8	0060520515	The Other Side Of The Story: A Novel	Marian Keyes	2, 4, 5
9	0060557257	Sushi For Beginners	Marian Keyes	1, 3
10	0060916508	Their Eyes Were Watching God: A Novel	Zora Neale Hurston	1, 3, 5
11	0060931418	Their Eyes Were Watching God	Zora Neale Hurston	1, 3, 5
12	006104055X	A Man To Die For	Eileen Dreyer	1, 3, 5

- SQL Search when input is Author Name or substring of Author Name

```sql
SELECT f.isbn, f.title, f.authors, f.branch_availability FROM (
SELECT t.isbn AS isbn, b.title AS title, t.authors AS authors,
book_copies.available_at_branch AS branch_availability,
t.isbn||b.title||t.authors AS search_string FROM project1_books b
INNER JOIN (SELECT ba.isbn AS isbn, LISTAGG(DISTINCT(a.name), ', ') AS
authors FROM project1_book_authors ba
INNER JOIN project1_authors a ON ba.author_id = a.author_id
GROUP BY ba.isbn) t ON b.isbn = t.isbn
INNER JOIN (SELECT isbn, LISTAGG(DISTINCT(branch_id), ', ') AS
available_at_branch FROM project1_book_copies GROUP BY isbn) book_copies ON
book_copies.isbn = b.isbn) f
WHERE LOWER(f.search_string) LIKE '%nath%'; --INPUT
```



```sql
SELECT f.isbn, f.title, f.authors, f.branch_availability FROM (
SELECT t.isbn AS isbn, b.title AS title, t.authors AS authors, book_copies.available_at_branch AS branch_availability, t.isbn||b.title||t.authors AS search_string FROM project1_books b
INNER JOIN (SELECT ba.isbn AS isbn, LISTAGG(DISTINCT(a.name), ', ') AS authors FROM project1_book_authors ba
INNER JOIN project1_authors a ON ba.author_id = a.author_id
GROUP BY ba.isbn) t ON b.isbn = t.isbn
INNER JOIN (SELECT isbn, LISTAGG(DISTINCT(branch_id), ', ') AS available_at_branch FROM project1_book_copies GROUP BY isbn) book_copies ON book_copies.isbn = b.isbn) f
WHERE LOWER(f.search_string) LIKE '%nath%'; --INPUT
```

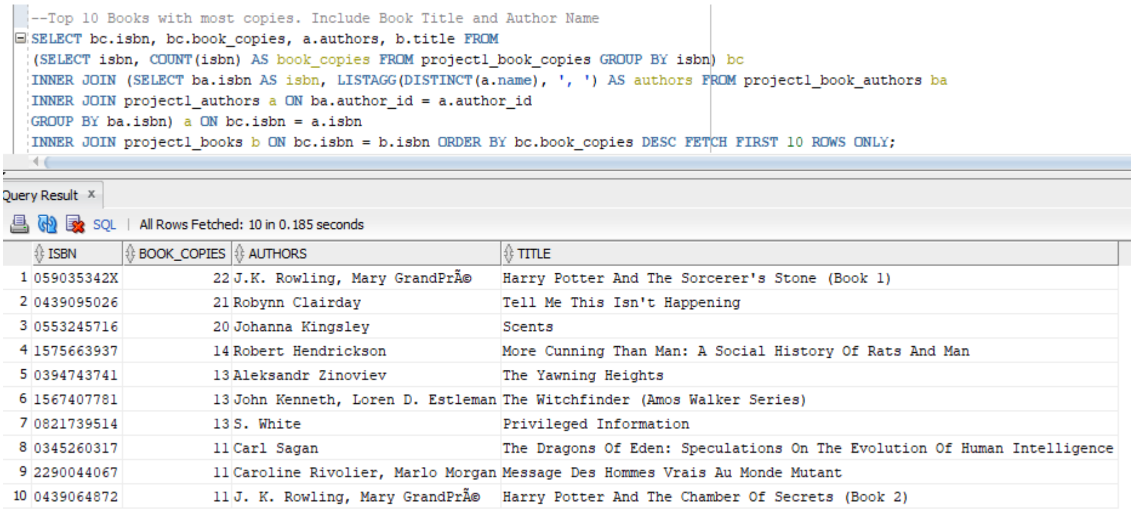
Fetched 50 rows in 0.323 seconds

	ISBN	TITLE	AUTHORS	BRANCH_AVAILABILITY
1	0060529709	Everything Is Illuminated: A Novel	Jonathan Safran Foer	3, 4
2	0060956453	Ordinary Resurrections: Children In The Years Of Hope	Jonathan Kozol	1, 2, 5
3	0060974990	Savage Inequalities: Children In America's Schools	Jonathan Kozol	1, 2, 3, 4, 5
4	0060976977	Amazing Grace: The Lives Of Children And The Conscience Of A Nation	Jonathan Kozol	1, 2, 5
5	0064440257	Oscar Otter (I Can Read Book 1)	Arnold Lobel, Nathaniel Benchley	1, 2, 3, 4, 5
6	0064441075	Sam The Minuteman (I Can Read Book 3)	Arnold Lobel, Nathaniel Benchley	1, 3, 4, 5
7	0071364498	Gipsy Moth Circles The World (The Sailor's Classics #1)	Jonathan Raban, Sir Francis Chichester	1, 2, 3, 5
8	0140069704	The Sleepers Of Erin	Jonathan Gash	1, 3, 5
9	0140106464	MOON SPENDER	Jonathan Gash	1, 2, 4, 5
10	0140147381	The Very Last Gambado (Lovejoy Mystery)	Jonathan Gash	1, 3, 4, 5
11	014062080X	Scarlet Letter (Penguin Popular Classics)	Nathaniel Hawthorne	1, 3, 5
12	0140620842	Gulliver's Travels (Penguin Popular Classics)	Jonathan Swift	1, 2, 3, 4, 5

Reports

1. ***Top 10 Books with most copies. Include Book Title and Author Name***

```
SELECT bc.isbn, bc.book_copies, a.authors, b.title FROM
(SELECT isbn, COUNT(isbn) AS book_copies FROM project1_book_copies GROUP BY
isbn) bc
INNER JOIN (SELECT ba.isbn AS isbn, LISTAGG(DISTINCT(a.name), ', ') AS
authors FROM project1_book_authors ba
INNER JOIN project1_authors a ON ba.author_id = a.author_id
GROUP BY ba.isbn) a ON bc.isbn = a.isbn
INNER JOIN project1_books b ON bc.isbn = b.isbn ORDER BY bc.book_copies DESC
FETCH FIRST 10 ROWS ONLY;
```



```
--Top 10 Books with most copies. Include Book Title and Author Name
SELECT bc.isbn, bc.book_copies, a.authors, b.title FROM
(SELECT isbn, COUNT(isbn) AS book_copies FROM project1_book_copies GROUP BY isbn) bc
INNER JOIN (SELECT ba.isbn AS isbn, LISTAGG(DISTINCT(a.name), ', ') AS authors FROM project1_book_authors ba
INNER JOIN project1_authors a ON ba.author_id = a.author_id
GROUP BY ba.isbn) a ON bc.isbn = a.isbn
INNER JOIN project1_books b ON bc.isbn = b.isbn ORDER BY bc.book_copies DESC FETCH FIRST 10 ROWS ONLY;
```

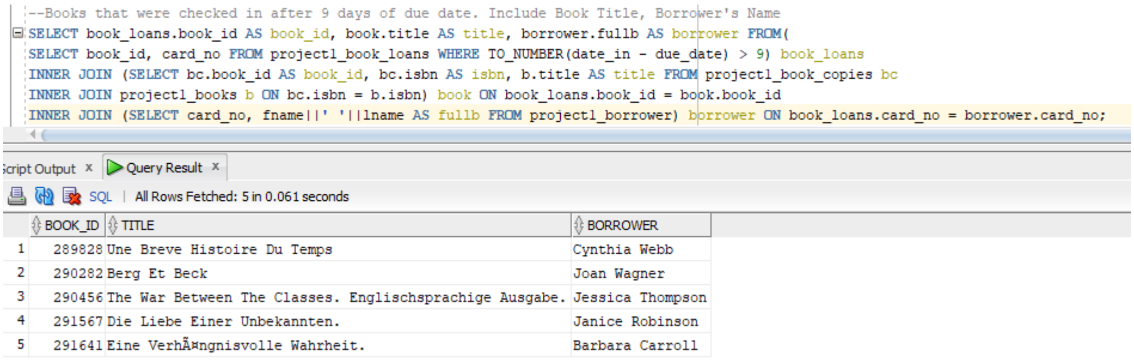
Query Result ×

SQL | All Rows Fetched: 10 in 0.185 seconds

	ISBN	BOOK_COPIES	AUTHORS	TITLE
1	059035342X	22	J.K. Rowling, Mary GrandPrÃ©	Harry Potter And The Sorcerer's Stone (Book 1)
2	0439095026	21	Robynn Clairday	Tell Me This Isn't Happening
3	0553245716	20	Johanna Kingsley	Scents
4	1575663937	14	Robert Hendrickson	More Cunning Than Man: A Social History Of Rats And Man
5	0394743741	13	Aleksandr Zinoviev	The Yawning Heights
6	1567407781	13	John Kenneth, Loren D. Estleman	The Witchfinder (Amos Walker Series)
7	0821739514	13	S. White	Privileged Information
8	0345260317	11	Carl Sagan	The Dragons Of Eden: Speculations On The Evolution Of Human Intelligence
9	2290044067	11	Caroline Rivolier, Marlo Morgan	Message Des Hommes Vrais Au Monde Mutant
10	0439064872	11	J. K. Rowling, Mary GrandPrÃ©	Harry Potter And The Chamber Of Secrets (Book 2)

2. ***Books that were checked in after 9 days of due date. Include Book Title, Borrower's Name***

```
SELECT book_loans.book_id AS book_id, book.title AS title, borrower.fullb AS
borrower FROM(
SELECT book_id, card_no FROM project1_book_loans WHERE TO_NUMBER(date_in -
due_date) > 9) book_loans
INNER JOIN (SELECT bc.book_id AS book_id, bc.isbn AS isbn, b.title AS title
FROM project1_book_copies bc
INNER JOIN project1_books b ON bc.isbn = b.isbn) book ON book_loans.book_id
= book.book_id
INNER JOIN (SELECT card_no, fname||' '||lname AS fullb FROM
project1_borrower) borrower ON book_loans.card_no = borrower.card_no;
```



```
--Books that were checked in after 9 days of due date. Include Book Title, Borrower's Name
SELECT book_loans.book_id AS book_id, book.title AS title, borrower.fullb AS borrower FROM(
SELECT book_id, card_no FROM project1_book_loans WHERE TO_NUMBER(date_in - due_date) > 9) book_loans
INNER JOIN (SELECT bc.book_id AS book_id, bc.isbn AS isbn, b.title AS title FROM project1_book_copies bc
INNER JOIN project1_books b ON bc.isbn = b.isbn) book ON book_loans.book_id = book.book_id
INNER JOIN (SELECT card_no, fname||' '||lname AS fullb FROM project1_borrower) borrower ON book_loans.card_no = borrower.card_no;
```

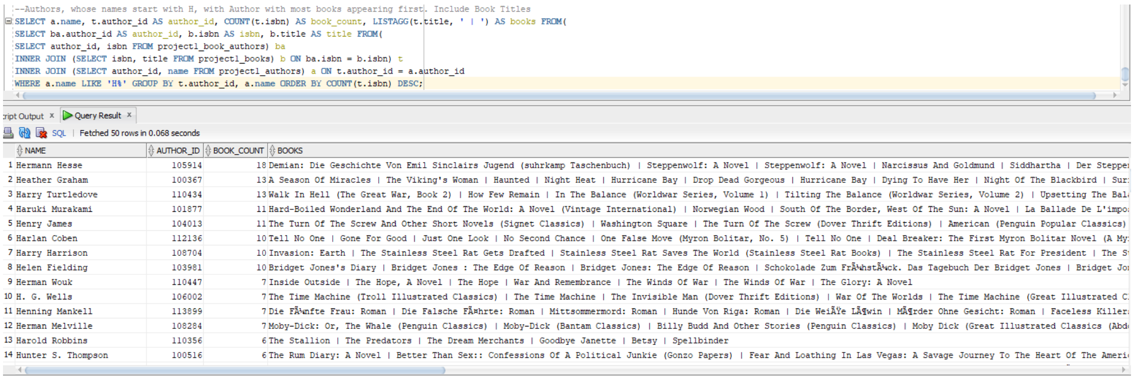
Script Output × Query Result ×

SQL | All Rows Fetched: 5 in 0.061 seconds

	BOOK_ID	TITLE	BORROWER
1	289828	Une Breve Histoire Du Temps	Cynthia Webb
2	290282	Berg Et Beck	Joan Wagner
3	290456	The War Between The Classes. Englischsprachige Ausgabe.	Jessica Thompson
4	291567	Die Liebe Einer Unbekannten.	Janice Robinson
5	291641	Eine VerhÃ¤ngnisvolle Wahrheit.	Barbara Carroll

3. ***Authors, whose names start with H, with Author with most books appearing first. Include Book Titles***

```
SELECT a.name, t.author_id AS author_id, COUNT(t.isbn) AS book_count,
LISTAGG(t.title, ' | ') AS books FROM(
SELECT ba.author_id AS author_id, b.isbn AS isbn, b.title AS title FROM(
SELECT author_id, isbn FROM project1_book_authors) ba
INNER JOIN (SELECT isbn, title FROM project1_books) b ON ba.isbn = b.isbn) t
INNER JOIN (SELECT author_id, name FROM project1_authors) a ON t.author_id =
a.author_id
WHERE a.name LIKE 'H%' GROUP BY t.author_id, a.name ORDER BY COUNT(t.isbn)
DESC;
```



```
--Authors, whose names start with H, with Author with most books appearing first. Include Book Titles
SELECT a.name, t.author_id AS author_id, COUNT(t.isbn) AS book_count, LISTAGG(t.title, ' | ') AS books FROM(
SELECT ba.author_id AS author_id, b.isbn AS isbn, b.title AS title FROM(
SELECT author_id, isbn FROM project1_book_authors) ba
INNER JOIN (SELECT isbn, title FROM project1_books) b ON ba.isbn = b.isbn) t
INNER JOIN (SELECT author_id, name FROM project1_authors) a ON t.author_id = a.author_id
WHERE a.name LIKE 'H%' GROUP BY t.author_id, a.name ORDER BY COUNT(t.isbn) DESC;
```

Fetched 50 rows in 0.068 seconds

	NAME	AUTHOR_ID	BOOK_COUNT	BOOKS
1	Hermann Hesse	105914	18	Demian: Die Geschichte Von Emil Sinclairs Jugend (suhrkamp Taschenbuch) | Steppenwolf: A Novel | Steppenwolf: A Novel | Narcissus And Goldmund | Siddhartha | Der Steppe
2	Heather Graham	100367	13	A Season Of Miracles | The Viking's Woman | Haunted | Night Heat | Hurricane Bay | Drop Dead Gorgeous | Hurricane Bay | Dying To Have Her | Night Of The Blackbird | Sur
3	Harry Turtledove	110434	13	Walk In Hell (The Great War, Book 2) | How Few Remain | In The Balance (Worldwar Series, Volume 1) | Tilting The Balance (Worldwar Series, Volume 2) | Upsetting The Bal
4	Haruki Murakami	101877	11	Hard-Boiled Wonderland And The End Of The World: A Novel (Vintage International) | Norwegian Wood | South Of The Border, West Of The Sun: A Novel | La Ballade De L'impo
5	Henry James	104013	11	The Turn Of The Screw And Other Short Novels (Signet Classics) | Washington Square | The Turn Of The Screw (Dover Thrift Editions) | American (Penguin Popular Classics)
6	Harlan Coben	112136	10	Tell No One | Gone For Good | Just One Look | No Second Chance | One False Move (Myron Bolitar, No. 5) | Tell No One | Deal Breaker: The First Myron Bolitar Novel (A My
7	Harry Harrison	108704	10	Invasion: Earth | The Stainless Steel Rat Gets Drafted | Stainless Steel Rat Saves The World (Stainless Steel Rat Books) | The Stainless Steel Rat For President | The S
8	Helen Fielding	103981	10	Bridget Jones's Diary | Bridget Jones : The Edge Of Reason | Bridget Jones: The Edge Of Reason | Schokolade Zum FrÃ¼hstÃ¼ck. Das Tagebuch Der Bridget Jones | Bridget Jo
9	Herman Wouk	110447	7	Inside Outside | The Hope, A Novel | The Hope | War And Remembrance | The Winds Of War | The Winds Of War | The Glory: A Novel
10	H. G. Wells	106002	7	The Time Machine (Troll Illustrated Classics) | The Time Machine | The Invisible Man (Dover Thrift Editions) | War Of The Worlds | The Time Machine (Great Illustrated C
11	Henning Mankell	113899	7	Die FÃ¼nfte Frau: Roman | Die Falsche FÃ¤hrte: Roman | Mittsommermord: Roman | Hunde Von Riga: Roman | Die WeiÃŸe LÃ¶win | MÃ¶rder Ohne Gesicht: Roman | Faceless Killer
12	Herman Melville	108284	7	Moby-Dick: Or, The Whale (Penguin Classics) | Moby-Dick (Bantam Classics) | Billy Budd And Other Stories (Penguin Classics) | Moby Dick (Great Illustrated Classics (Abd
13	Harold Robbins	110356	6	The Stallion | The Predators | The Dream Merchants | Goodbye Janette | Betsy | Spellbinder
14	Hunter S. Thompson	100516	6	The Rum Diary: A Novel | Better Than Sex:: Confessions Of A Political Junkie (Gonzo Papers) | Fear And Loathing In Las Vegas: A Savage Journey To The Heart Of The Ameri