Project Overview

The project will involve creating a database schema for a library management system that tracks books, authors, publishers, borrowers, and loans. It will also involve writing PL/SQL code to create tables, insert sample data, write queries, and create views, functions, triggers, and procedures.

1. Table Creation

Here are the table specifications for the library management system:

Table: Authors

- Columns:
  - `author_id` (number, primary key)
  - `first_name` (varchar2, 50)
  - `last_name` (varchar2, 50)
  - `date_of_birth` (date)

Table: Publishers

- Columns:
  - `publisher_id` (number, primary key)
  - `name` (varchar2, 100)
  - `address` (varchar2, 200)

Table: Books

- Columns:
  - `book_id` (number, primary key)
  - `title` (varchar2, 200)
  - `author_id` (number, foreign key to `authors.author_id`)
  - `publisher_id` (number, foreign key to `publishers.publisher_id`)
  - `published_date` (date)

Table: Borrowers

- Columns:
  - `borrower_id` (number, primary key)

- `first_name` (varchar2, 50)
- `last_name` (varchar2, 50)
- `address` (varchar2, 200)
- `phone_number` (varchar2, 20)

Table: Loans

- Columns:
  - `loan_id` (number, primary key)
  - `book_id` (number, foreign key to `books.book_id`)
  - `borrower_id` (number, foreign key to `borrowers.borrower_id`)
  - `loan_date` (date)
  - `return_date` (date)

2. Sample Data

Insert sample data in all tables.

3. Query

Write a query that retrieves the title and author name of all books borrowed by a specific borrower

4. View

Create a view that displays the loan history of a specific borrower

5. Function

Create a function that retrieves the number of books borrowed by a specific borrower

6. Trigger

Create a trigger that logs all loan transactions. Hint: You must create a ‘transaction_log’ table for this.

7. Procedure

Create a procedure that allows borrowers to return books

‘

Solutions

1. Table Creation

```
-- create table authors
CREATE TABLE authors (
  author_id NUMBER PRIMARY KEY,
  first_name VARCHAR2(50),
  last_name VARCHAR2(50),
  date_of_birth DATE
);

-- create table publishers
CREATE TABLE publishers (
  publisher_id NUMBER PRIMARY KEY,
  name VARCHAR2(100),
  address VARCHAR2(200)
);

-- create table books
CREATE TABLE books (
  book_id NUMBER PRIMARY KEY,
  title VARCHAR2(200),
  author_id NUMBER,
  publisher_id NUMBER,
  published_date DATE,
  FOREIGN KEY (author_id) REFERENCES authors(author_id),
  FOREIGN KEY (publisher_id) REFERENCES publishers(publisher_id)
);

-- create table borrowers
CREATE TABLE borrowers (
  borrower_id NUMBER PRIMARY KEY,
  first_name VARCHAR2(50),
  last_name VARCHAR2(50),
  address VARCHAR2(200),
  phone_number VARCHAR2(20)
);
```

```
-- create table loans
CREATE TABLE loans (
  loan_id NUMBER PRIMARY KEY,
  book_id NUMBER,
  borrower_id NUMBER,
  loan_date DATE,
  return_date DATE,
  FOREIGN KEY (book_id) REFERENCES books(book_id),
  FOREIGN KEY (borrower_id) REFERENCES borrowers(borrower_id)
);
```

2. Table Creation

```
-- insert sample data into authors table
INSERT INTO authors (author_id, first_name, last_name, date_of_birth)
VALUES (1, 'J.K.', 'Rowling', TO_DATE('31-JUL-1965', 'DD-MON-YYYY'));

INSERT INTO authors (author_id, first_name, last_name, date_of_birth)
VALUES (2, 'Stephen', 'King', TO_DATE('21-SEP-1947', 'DD-MON-YYYY'));

-- insert sample data into publishers table
INSERT INTO publishers (publisher_id, name, address)
VALUES (1, 'Scholastic', '557 Broadway, New York, NY 10012');

INSERT INTO publishers (publisher_id, name, address)
VALUES (2, 'Doubleday', '1745 Broadway, New York, NY 10019');

-- insert sample data into books table
INSERT INTO books (book_id, title, author_id, publisher_id, published_date)
VALUES (1, 'Harry Potter and the Philosopher''s Stone', 1, 1, TO_DATE('26-JUN-1997',
'DD-MON-YYYY'));

INSERT INTO books (book_id, title, author_id, publisher_id, published_date)
VALUES (2, 'The Shining', 2, 2, TO_DATE('28-JAN-1977', 'DD-MON-YYYY'));

-- insert sample data into borrowers table
```

```
INSERT INTO borrowers (borrower_id, first_name, last_name, address, phone_number)
VALUES (1, 'John', 'Doe', '123 Main St., Anytown, USA', '555-1234');

INSERT INTO borrowers (borrower_id, first_name, last_name, address, phone_number)
VALUES (2, 'Jane', 'Doe', '456 Oak St., Anytown, USA', '555-5678');

-- insert sample data into loans table
INSERT INTO loans (loan_id, book_id, borrower_id, loan_date, return_date)
VALUES (1, 1, 1, TO_DATE('01-JAN-2023', 'DD-MON-YYYY'), NULL);

INSERT INTO loans (loan_id, book_id, borrower_id, loan_date, return_date)
VALUES (2, 2, 2, TO_DATE('02-JAN-2023', 'DD-MON-YYYY'), TO_DATE('10-JAN-2023',
'DD-MON-YYYY'));
```

3. Query

```
SELECT b.title, CONCAT(a.first_name, ' ', a.last_name) AS author_name
FROM books b
JOIN authors a ON b.author_id = a.author_id
JOIN loans l ON b.book_id = l.book_id
JOIN borrowers br ON l.borrower_id = br.borrower_id
WHERE br.borrower_id = 1;
```

4. View

```
CREATE VIEW borrower_loans AS
SELECT l.loan_id, b.title, CONCAT(a.first_name, ' ', a.last_name) AS author_name,
l.loan_date, l.return_date
FROM books b
JOIN authors a ON b.author_id = a.author_id
JOIN loans l ON b.book_id = l.book_id
JOIN borrowers br ON l.borrower_id = br.borrower_id
WHERE br.borrower_id = 1;
```

5. Function

```
CREATE FUNCTION get_borrowed_books_count(p_borrower_id IN NUMBER)
RETURN NUMBER
IS
  l_count NUMBER;
BEGIN
  SELECT COUNT(*)
  INTO l_count
  FROM loans
  WHERE borrower_id = p_borrower_id;

  RETURN l_count;
END;
```

6. Trigger

```
CREATE OR REPLACE TRIGGER log_loan_transaction
AFTER INSERT OR UPDATE ON loans
FOR EACH ROW
DECLARE
  l_log_text VARCHAR2(200);
BEGIN
  l_log_text := 'Transaction: ';
  IF INSERTING THEN
    l_log_text := l_log_text || 'Insert, ';
  ELSE
    l_log_text := l_log_text || 'Update, ';
  END IF;

  l_log_text := l_log_text || 'Book ID: ' || :NEW.book_id || ', ';
  l_log_text := l_log_text || 'Borrower ID: ' || :NEW.borrower_id || ', ';
  l_log_text := l_log_text || 'Loan Date: ' || TO_CHAR(:NEW.loan_date, 'DD-MON-YYYY') || ',
';

  IF :NEW.return_date IS NOT NULL THEN
```

```
    l_log_text := l_log_text || 'Return Date: ' || TO_CHAR(:NEW.return_date, 'DD-MON-
YYYY');
  END IF;

  INSERT INTO transaction_log (log_text)
  VALUES (l_log_text);
END;
```

7. Procedure

```
CREATE OR REPLACE PROCEDURE return_book(p_loan_id IN NUMBER, p_return_date IN
DATE)
IS
  l_book_id NUMBER;
BEGIN
  -- update loan record
  UPDATE loans
  SET return_date = p_return_date
  WHERE loan_id = p_loan_id;

  -- update book record
  SELECT book_id INTO l_book_id
  FROM loans
  WHERE loan_id = p_loan_id;

  UPDATE books
  SET available = 'Y'
  WHERE book_id = l_book_id;

  COMMIT;
END;
```