COMP 3610 : Database Systems
Winter 2023
Activity #6– Stored Procedures/functions

Objective:
To add automatic calculations of the fine.

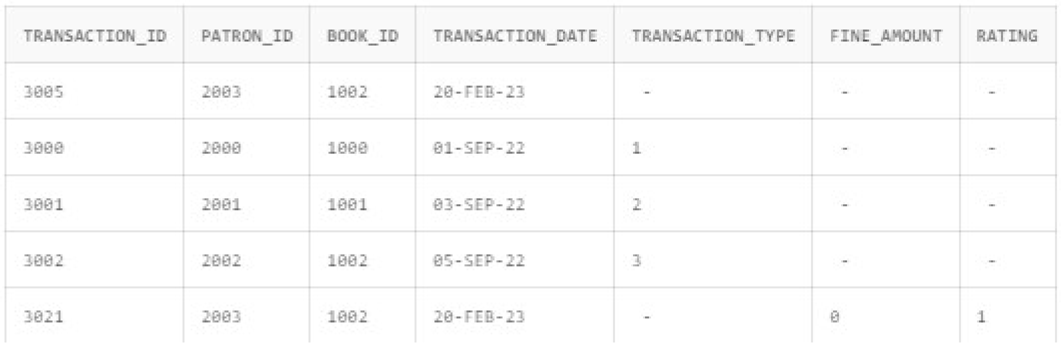
In the schema used for your assignment#1 – (books, Patrons and Transactions), you also added a column named as fine (numeric) in the transaction table. This is for storing the amount of fine paid by the patron when a book is returned late.

To do:
Q1. [10 marks]
Create a Procedure issueBook, where patron_id and book_id is taken as parameters.

issueBook(patron_id, book_id)
This function will run when the patron is coming to issue a book.
Steps:
Verify the patron_id in the patron table
Verify the book_id in the book table (not a reference book)
Insert the new row in transaction tale with
-patronid, book id, transactiondate as today, fine – 0 and rating as 1.

The transaction_id should automatically generate using sequence

```
CREATE OR REPLACE PROCEDURE issueBook (
  p_patron_id    patrons.patron_id%TYPE,
  p_book_id      books.book_id%TYPE
) AS
  patron_exists INT;
  book_exists INT;
  transaction_id INT;
  today_date DATE;
BEGIN
  -- for patron
  SELECT COUNT(*) INTO patron_exists FROM patrons WHERE patron_id =
p_patron_id;   -- check patron exists or not in table
  IF patron_exists = 0 THEN -- raise an error if patron id is invalid
    RAISE_APPLICATION_ERROR(-20001, 'you entered an invalid patron ID ');
  END IF;

  -- for book
  SELECT COUNT(*) INTO book_exists FROM books WHERE book_id = p_book_id
AND rating != 1;
  IF book_exists = 0 THEN
    RAISE_APPLICATION_ERROR(-20002, 'you entered an invalid book ID ');
  END IF;

  transaction_id := sequence_transaction.nextval;
  today_date := sysdate;
  INSERT INTO transactions (transaction_id , patron_id , book_id  ,
transaction_date,FINE_AMOUNT,rating)
  VALUES (sequence_transaction.nextval,p_patron_id,p_book_id, today_date,0,'1');

END;

EXEC issueBook(2003,1002 );
```



TRANSACTION_ID	PATRON_ID	BOOK_ID	TRANSACTION_DATE	TRANSACTION_TYPE	FINE_AMOUNT	RATING
3005	2003	1002	20-FEB-23	-	-	-
3000	2000	1000	01-SEP-22	1	-	-
3001	2001	1001	03-SEP-22	2	-	-
3002	2002	1002	05-SEP-22	3	-	-
3021	2003	1002	20-FEB-23	-	0	1

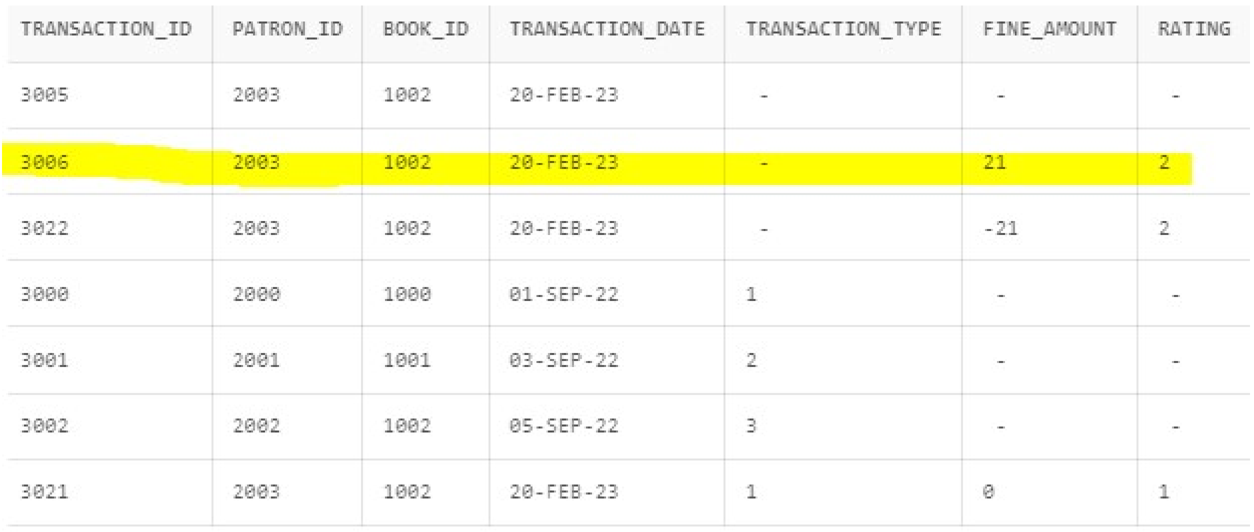
Q2. Create another procedure [10 marks]
returnBook(patron-Id, book_id)

Create a procedure that should take in patron_id and book_id as parameters. The procedure will calculate the amount of fine due on the book.

This procedure will run when the patron is coming back to return the book.
Steps:
Read the tuple with the patron_id and book_id with type=1. Check the transaction date.
Add 7 to that. If todays date > calculated date, calculate the fine.
Now insert the new row in transaction tale with
-patronid, book id, transactiondate as today, fine – as calculated and rating as 2.

The transaction_id should automatically generate using sequence.

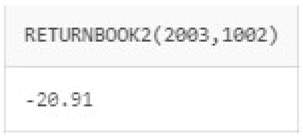
```
CREATE OR REPLACE PROCEDURE returnBook (
  p_patron_id    patrons.patron_id%TYPE,
  p_book_id      books.book_id%TYPE
) AS
   extract_date DATE;
   fine         NUMBER(8,2);
BEGIN
  SELECT TRANSACTION_DATE INTO extract_date FROM transactions WHERE
TRANSACTION_TYPE = '1' AND  PATRON_ID = p_patron_id AND BOOK_ID =
p_book_id;
  extract_date := extract_date + INTERVAL '7' DAY;
  IF sysdate < extract_date THEN
    fine := (SYSDATE - extract_date) * 3; -- 3 dollar per day fine
    DBMS_OUTPUT.PUT_LINE(fine);

    INSERT INTO transactions (transaction_id , patron_id , book_id  ,
transaction_date,FINE_AMOUNT,rating)
    VALUES (sequence_transaction.nextval,p_patron_id,p_book_id,
sysdate,ABS(fine),'2');

  END IF;
END;

EXEC returnBook(2003,1002);
SELECT * FROM transactions;
```



TRANSACTION_ID	PATRON_ID	BOOK_ID	TRANSACTION_DATE	TRANSACTION_TYPE	FINE_AMOUNT	RATING
3005	2003	1002	20-FEB-23	-	-	-
3006	2003	1002	20-FEB-23	-	21	2
3022	2003	1002	20-FEB-23	-	-21	2
3000	2000	1000	01-SEP-22	1	-	-
3001	2001	1001	03-SEP-22	2	-	-
3002	2002	1002	05-SEP-22	3	-	-
3021	2003	1002	20-FEB-23	1	0	1

Q3. Convert returnBook into a function that will return the fine amount. [5 marks]

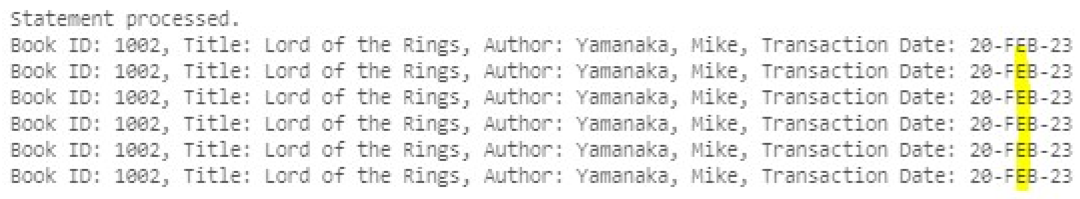
```
CREATE OR REPLACE FUNCTION returnBook2 (
  p_patron_id    patrons.patron_id%TYPE,
  p_book_id      books.book_id%TYPE
) RETURN NUMBER
AS
   extract_date DATE;
   fine         NUMBER(8,2);
BEGIN
  SELECT TRANSACTION_DATE INTO extract_date FROM transactions WHERE
TRANSACTION_TYPE = '1' AND  PATRON_ID = p_patron_id AND BOOK_ID = p_book_id;
  extract_date := extract_date + INTERVAL '7' DAY;
  IF sysdate < extract_date THEN
    fine := (SYSDATE - extract_date) * 3; -- 3 dollar per day fine
    DBMS_OUTPUT.PUT_LINE(fine);
    RETURN fine;
  ELSE
    RETURN 0;
  END IF;

END;

SELECT returnBook2(2003,1002) FROM DUAL;
```



RETURNBOOK2(2003,1002)
-20.91

Q4. Create a procedure to display the details of all the books that have been issues in the month of February. [10 marks]

```
CREATE OR REPLACE PROCEDURE display_books_issued_in_feb
IS
 v_book_id books.book_id%TYPE;
 v_title books.title%TYPE;
 v_author_last_name books.author_last_name%TYPE;
 v_author_first_name books.author_first_name%TYPE;
 v_transaction_date transactions.transaction_date%TYPE;

 CURSOR c_books_issued_in_feb IS
  SELECT b.book_id, b.title, b.author_last_name, b.author_first_name, t.transaction_date
  FROM books b
  JOIN transactions t ON b.book_id = t.book_id
  WHERE TO_CHAR(t.transaction_date, 'MM') = '02';
BEGIN

 OPEN c_books_issued_in_feb;
 LOOP
  FETCH c_books_issued_in_feb INTO v_book_id, v_title, v_author_last_name,
v_author_first_name, v_transaction_date;
  EXIT WHEN c_books_issued_in_feb%NOTFOUND;
  DBMS_OUTPUT.PUT_LINE('Book ID: ' || v_book_id || ', Title: ' || v_title || ', Author: ' ||
v_author_last_name || ', ' || v_author_first_name || ', Transaction Date: ' ||
v_transaction_date);
 END LOOP;
 CLOSE c_books_issued_in_feb;

END;

INSERT INTO transactions (transaction_id , patron_id , book_id  , transaction_date  ,
transaction_type)
VALUES (sequence_transaction.nextval,2002,1004, to_date('02-05-22','MM-DD-YY') ,'1');
SELECT * FROM transactions;

EXEC display_books_issued_in_feb;
```



```
Statement processed.
Book ID: 1002, Title: Lord of the Rings, Author: Yamanaka, Mike, Transaction Date: 20-FEB-23
Book ID: 1002, Title: Lord of the Rings, Author: Yamanaka, Mike, Transaction Date: 20-FEB-23
Book ID: 1002, Title: Lord of the Rings, Author: Yamanaka, Mike, Transaction Date: 20-FEB-23
Book ID: 1002, Title: Lord of the Rings, Author: Yamanaka, Mike, Transaction Date: 20-FEB-23
Book ID: 1002, Title: Lord of the Rings, Author: Yamanaka, Mike, Transaction Date: 20-FEB-23
Book ID: 1002, Title: Lord of the Rings, Author: Yamanaka, Mike, Transaction Date: 20-FEB-23
```