

Unnamed PL/SQL code block: Use of Control structure and Exception handling is mandatory.

Suggested Problem statement: Consider Tables:

1. Borrower(Roll_no, Name, DateofIssue, NameofBook, Status)
2. Fine(Roll_no,Date,Amt)

- Accept Roll_no and NameofBook from user.
- Check the number of days (from date of issue).
- If days are between 15 to 30 then fine amount will be Rs 5per day.
- If no. of days>30, per day fine will be Rs 50 per day and for days less than 30, Rs. 5 per day.
- After submitting the book, status will change from I to R.
- If condition of fine is true, then details will be stored into fine table.
- Also handles the exception by named exception handler or user define exception handler

PL/SQL Code Block

sql

Copy code

DECLARE

-- Declare variables to hold input values

v_roll_no Borrower.Roll_no%TYPE;

v_name_of_book Borrower.NameofBook%TYPE;

-- Declare variables for fine calculation

v_date_of_issue Borrower.DateofIssue%TYPE;

v_current_date DATE := SYSDATE;

v_days_overdue NUMBER;

v_fine_amount NUMBER := 0;

v_status Borrower.Status%TYPE;

-- Exception handling variables

v_no_book_found EXCEPTION;

v_invalid_roll_no EXCEPTION;

BEGIN

-- Accept input from the user

v_roll_no := &roll_no; -- e.g. 101

```

v_name_of_book := '&name_of_book'; -- e.g. 'Database Management'

-- Retrieve the date of issue and current status from Borrower table
SELECT DateofIssue, Status INTO v_date_of_issue, v_status
FROM Borrower
WHERE Roll_no = v_roll_no AND NameofBook = v_name_of_book;

-- Check if the book was issued
IF v_status = 'I' THEN
    -- Calculate number of days overdue
    v_days_overdue := TRUNC(v_current_date - v_date_of_issue);

    -- Calculate fine based on overdue days
    IF v_days_overdue BETWEEN 15 AND 30 THEN
        v_fine_amount := v_days_overdue * 5; -- Rs 5 per day
    ELSIF v_days_overdue > 30 THEN
        v_fine_amount := (30 * 5) + ((v_days_overdue - 30) * 50); -- Rs 5 for first 30 days, Rs 50 for
later
    END IF;

    -- Update status of the book to Returned
    UPDATE Borrower
    SET Status = 'R'
    WHERE Roll_no = v_roll_no AND NameofBook = v_name_of_book;

    -- If fine amount is greater than 0, insert fine details into Fine table
    IF v_fine_amount > 0 THEN
        INSERT INTO Fine (Roll_no, Date, Amt)
        VALUES (v_roll_no, v_current_date, v_fine_amount);
        DBMS_OUTPUT.PUT_LINE('Fine recorded: Rs ' || v_fine_amount);
    ELSE

```

```

        DBMS_OUTPUT.PUT_LINE('No fine applicable. Book returned on time.');
```

END IF;


```

ELSE

    RAISE v_no_book_found; -- Raise exception if the book is not found or already returned
END IF;

EXCEPTION

    WHEN v_no_book_found THEN

        DBMS_OUTPUT.PUT_LINE('No record found for the given Roll Number and Book Name.');
```

WHEN NO_DATA_FOUND THEN

```

        DBMS_OUTPUT.PUT_LINE('Invalid Roll Number or Book Name.');
```

WHEN OTHERS THEN

```

        DBMS_OUTPUT.PUT_LINE('An unexpected error occurred: ' || SQLERRM);
END;
```

/

Explanation of the Code

1. Variable Declarations:

- Variables are declared for roll number, book name, date of issue, current date, days overdue, and fine amount.
- Two user-defined exceptions (v_no_book_found and v_invalid_roll_no) are declared for handling specific errors.

2. Input Handling:

- User input is accepted for Roll_no and NameofBook.

3. Select Query:

- A query retrieves the date of issue and current status for the given roll number and book name.

4. Control Structures:

- An IF statement checks if the status of the book is 'I' (Issued).
- The number of days overdue is calculated and appropriate fines are calculated based on the specified conditions.

5. Update and Insert:

- If the book is returned and there is a fine, the fine details are inserted into the Fine table.
- The status of the book is updated to 'R' (Returned).

6. Exception Handling:

- A WHEN clause is used to handle specific exceptions:
 - v_no_book_found for cases where the book was not found.
 - NO_DATA_FOUND for invalid roll numbers or book names.
 - OTHERS for any unexpected errors, displaying a generic error message.

Usage

- To run this code, you need to be in an Oracle environment with access to the Borrower and Fine tables.
- Make sure to have the required privileges to perform the operations defined in the PL/SQL block.

This code ensures proper handling of control flow and exceptions while fulfilling the requirements specified in the problem statement.

2/2

4o mini

ANS=