**3: write a trigger for Library (bid, bname, doi, status) to update the number of copies (noc) according to ISSUE & RETURN status on update or insert query. Increase the noc if status is RETURN, Decrease noc if status is ISSUE in Library\_Audit table(bid,bname,noc,timestampofquery). Write a trigger after update on Library such that if doi is more than 20 days ago then status should be FINE and in the Library\_Audit table fine should be equal to no. of days \* 10.**

-- 1. Create Library table

CREATE TABLE Library (

bid NUMBER PRIMARY KEY,

bname VARCHAR2(50),

doi DATE,

status VARCHAR2(10),

noc NUMBER -- number of copies available

);

-- 2. Create Library\_Audit table

CREATE TABLE Library\_Audit (

audit\_id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

bid NUMBER,

bname VARCHAR2(50),

noc NUMBER,

timestampofquery DATE,

fine NUMBER

);

-- 3. Insert sample data into Library table

INSERT INTO Library VALUES (101, 'Mathematics', TO\_DATE('2025-04-10', 'YYYY-MM-DD'), 'ISSUE', 5);

INSERT INTO Library VALUES (102, 'Physics', TO\_DATE('2025-04-01', 'YYYY-MM-DD'), 'RETURN', 3);

INSERT INTO Library VALUES (103, 'Chemistry', TO\_DATE('2025-03-20', 'YYYY-MM-DD'), 'ISSUE', 7);

-- 4. Create trigger to update noc and insert audit record

CREATE OR REPLACE TRIGGER trg\_library\_update

AFTER INSERT OR UPDATE ON Library

FOR EACH ROW

DECLARE

v\_days NUMBER;

v\_fine NUMBER := NULL;

BEGIN

-- Calculate days since date of issue

v\_days := TRUNC(SYSDATE) - TRUNC(:NEW.doi);

-- Update number of copies based on status

IF INSERTING OR UPDATING THEN

IF :NEW.status = 'ISSUE' THEN

-- Decrease noc by 1

UPDATE Library SET noc = noc - 1 WHERE bid = :NEW.bid;

v\_fine := NULL; -- no fine on issue

ELSIF :NEW.status = 'RETURN' THEN

-- Increase noc by 1

UPDATE Library SET noc = noc + 1 WHERE bid = :NEW.bid;

v\_fine := NULL; -- no fine on return

END IF;

END IF;

-- If more than 20 days since issue, set status to FINE and calculate fine

IF UPDATING AND v\_days > 20 THEN

UPDATE Library SET status = 'FINE' WHERE bid = :NEW.bid;

-- Fine = (days overdue beyond 20) \* 10

v\_fine := (v\_days - 20) \* 10;

END IF;

-- Insert audit record with noc and fine info

INSERT INTO Library\_Audit (bid, bname, noc, timestampofquery, fine)

VALUES (:NEW.bid, :NEW.bname,

(SELECT noc FROM Library WHERE bid = :NEW.bid),

SYSDATE, v\_fine);

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Trigger error: ' || SQLERRM);

END;

/

**4: Write a database trigger on Library table. The System should keep track of the records that are being updated or deleted. The old value of updated or deleted records should be added in Library\_Audit table.**

-- 1. Create Library table

CREATE TABLE Library (

bid NUMBER PRIMARY KEY,

bname VARCHAR2(50),

doi DATE,

status VARCHAR2(10),

noc NUMBER

);

-- 2. Create Library\_Audit table to store old records on update/delete

CREATE TABLE Library\_Audit (

audit\_id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

bid NUMBER,

bname VARCHAR2(50),

doi DATE,

status VARCHAR2(10),

noc NUMBER,

action\_type VARCHAR2(10), -- 'UPDATE' or 'DELETE'

audit\_timestamp DATE

);

-- 3. Insert sample data into Library

INSERT INTO Library VALUES (101, 'Mathematics', TO\_DATE('2025-04-10', 'YYYY-MM-DD'), 'ISSUE', 5);

INSERT INTO Library VALUES (102, 'Physics', TO\_DATE('2025-04-01', 'YYYY-MM-DD'), 'RETURN', 3);

-- 4. Create trigger to log old records before UPDATE or DELETE

CREATE OR REPLACE TRIGGER trg\_library\_audit

BEFORE UPDATE OR DELETE ON Library

FOR EACH ROW

BEGIN

INSERT INTO Library\_Audit (

bid, bname, doi, status, noc, action\_type, audit\_timestamp

) VALUES (

:OLD.bid, :OLD.bname, :OLD.doi, :OLD.status, :OLD.noc,

CASE

WHEN UPDATING THEN 'UPDATE'

WHEN DELETING THEN 'DELETE'

END,

SYSDATE

);

END;

/