

Advanced Database Management System

Assignment : 2

Name: Hossain, MD Meraz

ID: 20-42460-1

Section: D

1. Suppose XYZ company has a rule stating that an employee's salary cannot be changed by more than 20% of the original salary if his/her experience is less than 5 years. Create a trigger 'salary_change_monitoring' to enforce this constraint. The trigger fires whenever there is an update to the emp table and outputs a suitable error message when the rule is violated.

Solution:

```
CREATE OR REPLACE TRIGGER salary_change_monitoring
BEFORE UPDATE OF SAL ON EMP
FOR EACH
ROW
DECLARE
    years_experience NUMBER;
BEGIN
    years_experience := EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
:OLD.HIREDATE);
    IF years_experience < 5 THEN
        IF :NEW.SAL > :OLD.SAL * 1.2 THEN
            RAISE_APPLICATION_ERROR(-20001, 'Salary cannot be changed by more than 20% of the
original salary if experience is less than 5 years');
        END
    IF;
END IF;
END;
```

2. Write database triggers that check that qty_on_hand does not become negative.

Solution:

```
CREATE OR REPLACE TRIGGER
check_qty_on_hand BEFORE UPDATE OF
QTY_ON_HND ON INVENTORY FOR EACH
ROW
DECLARE
E BEGIN
    IF :NEW.QTY_ON_HND < 0 THEN
        RAISE_APPLICATION_ERROR(-20002, 'Quantity on hand cannot become negative'); END
    IF;
```

END;

3. A trigger occurs when ever an insert or update occurs in bookshelf table and insert the old and new data in the bookshelf_audit table.

```
create table product_master (  
  pno varchar2(5), dsc  
  varchar2(10),  
  profit_percent number(3),  
  unit_measure varchar2(10),  
  qoh number(4), reorderlevel  
  number(8), sell_price  
  number(8), cose_price  
  number(8)  
);
```

```
create table bookshelf (  
  Title varchar2(6),  
  publisher varchar2(6),  
  rating varchar2(5)  
);  
create table bookshelf_audit (  
  Title varchar2(6),  
  publisher varchar2(6),  
  old_rating varchar2(5),  
  new_rating varchar2(5),  
  audit_dt date  
)
```

Solution:

```
CREATE OR REPLACE TRIGGER bookshelf_change_audit  
AFTER INSERT OR UPDATE ON bookshelf  
FOR EACH  
ROW BEGIN  
  IF UPDATING THEN  
    INSERT INTO bookshelf_audit (Title, publisher, old_rating, new_rating, audit_dt) VALUES (  
      :OLD.Title,  
      :OLD.publisher,  
      :OLD.rating,  
      :NEW.rating,
```

```

        SYSDATE
    );
ELSIF INSERTING THEN
    INSERT INTO bookself_audit (Title, publisher, old_rating, new_rating, audit_dt) VALUES (
        :NEW.Title,
        :NEW.publisher,
        NULL,
        :NEW.rating,
        SYSDATE
    );
END
IF;
END;

```

4. Create 2 tables, LAB7_RESERVATIONS table with 2 columns

```

Create table LAB7_RESERVATIONS
( flight_id char(6) not null,
  Customer_phone char(11) not null)

```

```

Createtable LAB7_flights (
  flight_id char(6) not null,
  Seats number(3) not null,
  Primarykey(flight_id))

```

```

Insertinto LAB7_flights values(AC0529,120)
Insertinto LAB7_flights values(AC0529,120)

```

Create a trigger LAB7_RES_TRG that will ensure that when a new row is inserted into the LAB7_RESERVATIONS table, the flight id is in the LAB7_FLIGHTS table and that the number of seats on this flight, SEATS is greater than 0. Here are the details of how the trigger should behave:

- If flight id is not in the flights table it should raise application error 'Invalid flight id'
- If flight id is in the flights table, (for example AC0529) but SEATS=0, then it should raise application error 'Flight AC0529 has no seats left'
- If flight id is in the flights table and SEATS>0, then it should update the appropriate row in flights table by setting SEATS=SEATS-1 for this flight.

Solution:

```

CREATE OR REPLACE TRIGGER
LAB7_RES_TRG BEFORE INSERT ON
LAB7_RESERVATIONS FOR EACH ROW
DECLARE
    v_seats LAB7_FLIGHTS.seats%TYPE;
BEGIN

```

```
SELECT seats INTO v_seats
FROM LAB7_FLIGHTS
WHERE flight_id = :NEW.flight_id;

IF v_seats IS NULL THEN
    RAISE_APPLICATION_ERROR(-20001, 'Invalid flight id');
ELSIF v_seats = 0 THEN
    RAISE_APPLICATION_ERROR(-20002, 'Flight'||:NEW.flight_id||'has no seats left'); ELSE
    UPDATE LAB7_FLIGHTS
    SET seats = seats - 1
    WHERE flight_id = :NEW.flight_id;
END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20001, 'Invalid flight id');
END;
```