Database Management System (LAB)

Assignment 12

Name: SHISHU
Reg. No.: 2020CA089

1. **Write a row trigger that copies the rows of 'Salary' table in a new table 'Salary_new' which has same schema. The trigger executes in case of updation of 'Salary' table. The schema for 'Salary' table is:** ***Salary(Emp_no, Basic, HRA, DA, Total_Dedution, Net_Salary, Gross_Salary)***

Create Table:

```
CREATE TABLE salary(emp_no number(2) primary key, basic_sal integer, da integer,
total_deduction integer, net_salary integer, gross_salary integer);

insert into salary values(2, 15000, 4000, 1000, 5000, 15000, 20000);

insert into salary values(1, 31000, 8000, 1000, 5000, 35000, 40000);

insert into salary values(3, 14000, 4000, 1000, 5000, 15000, 19000);

insert into salary values(4, 14000, 4000, 1000, 5000, 15000, 19000);

insert into salary values(5, 13000, 4000, 1000, 5000, 15000, 18000);
```

EMP_NO	BASIC_SAL	DA	TOTAL_DEDUCTION	NET_SALARY	GROSS_SALARY	HRA
2	15000	4000	1000	5000	15000	20000
5	13000	4000	1000	5000	15000	18000
1	31000	8000	1000	5000	35000	40000
3	14000	4000	1000	5000	15000	19000
4	14000	4000	1000	5000	15000	19000

5 rows returned in 0.02 seconds Download

Creating Trigger:

```
CREATE OR REPLACE TRIGGER salary_changes
BEFORE UPDATE ON salary
FOR EACH ROW
BEGIN
   INSERT INTO new_salary values(:OLD.emp_no , :OLD.basic_sal, :OLD.da,
:OLD.total_deduction, :OLD.net_salary, :OLD.gross_salary, :OLD.hra);
```

```
END;
```

```
Trigger created.

0.05 seconds
```

Quering:

```
UPDATE salary SET basic_sal = 20000 WHERE emp_no = 2;
```

Salary Table after modification:

EMP_NO	BASIC_SAL	DA	TOTAL_DEDUCTION	NET_SALARY	GROSS_SALARY	HRA
2	20000	4000	1000	5000	15000	20000
5	13000	4000	1000	5000	15000	18000
1	31000	8000	1000	5000	35000	40000
3	14000	4000	1000	5000	15000	19000
4	14000	4000	1000	5000	15000	19000

Row inserted after trigger called:

EMP_NO	BASIC_SAL	DA	TOTAL_DEDUCTION	NET_SALARY	GROSS_SALARY	HRA
2	15000	4000	1000	5000	15000	20000

1 rows returned in 0.02 seconds Download

2. **A Factory maintains records of Stock-On-Hand and material requirements in the Item_Master table andItem_Requisite table respectively. Write the code for trigger, which on deletion of any row in Item_Requisite, updates the Bal_Stock in the Item_Master table for that requisite ie it decreases theBal_Stock in the Item_Master table by the quantity in the Item_Requisite table. If the value in Bal_Stock becomes negative, the update operation should not be allowed.**

Table Creation:

```
CREATE TABLE item_master(item_id integer primary key, descr varchar(15), bal_stock integer);

INSERT into item_master values(101, 'keyboard', 80);
INSERT into item_master values(102, 'lathe_machine', 70);
INSERT into item_master values(103, 'compass', 100);
INSERT into item_master values(104, 'beaker', 90);
INSERT into item_master values(105, 'ammeter', 60);

CREATE TABLE item_requisite(item_id integer primary key, dep_code varchar(8), quantity integer);

INSERT into item_requisite values(101, 'cs', 30);
INSERT into item_requisite values(102, 'mech', 20);
INSERT into item_requisite values(103, 'civil', 25);
INSERT into item_requisite values(104, 'chem', 35);
INSERT into item_requisite values(105, 'elect', 80);
```

Table item_master :

ITEM_ID	DESCR	BAL_STOCK
105	ammeter	60
101	keyboard	80
103	compass	100
104	beaker	90
102	lathe_machine	70

5 rows returned in 0.01 seconds Download

Table item_requisite:

ITEM_ID	DEP_CODE	QUANTITY
101	cs	30
104	chem	35
103	civil	25
105	elect	80
102	mech	20

5 rows returned in 0.02 seconds Download

Trigger:

```
CREATE OR REPLACE TRIGGER factory_update
BEFORE DELETE ON item_requisite
FOR EACH ROW
DECLARE
    curr_stock item_master.bal_stock%type;
    insufficient_bal EXCEPTION;
    res integer;
BEGIN
   SELECT bal_stock into curr_stock from item_master where item_id = :OLD.item_id;
   if (curr_stock >= :OLD.quantity)
   then
        res := curr_stock - :OLD.quantity;
        UPDATE item_master set bal_stock = res where item_id = :OLD.item_id;
   else
        RAISE insufficient_bal;
   end if;
EXCEPTION
    WHEN insufficient_bal THEN
      dbms_output.put_line('Insufficient Stock Balance!!');
END;
```

Query:

```
DELETE from item_requisite where item_id = 104;
```

Table item_master:

ITEM_ID	DESCR	BAL_STOCK
105	ammeter	60
101	keyboard	80
103	compass	100
104	beaker	55
102	lathe_machine	70

5 rows returned in 0.01 seconds Download

Table item_requisite:

ITEM_ID	DEP_CODE	QUANTITY
101	cs	30
103	civil	25
105	elect	80
102	mech	20

4 rows returned in 0.01 seconds Download

---

3. **An HR system has an 'emp' table that holds a row for each employee in the company. Each record in the table has a manager field that holds name of the employee's manager. Write a trigger so that when a manager's record is deleted from the emp table, the mgr field of the employees working under that**

manager is set to NULL. In other words, implement the following SQL statement:

Create Table:

```
CREATE TABLE employee(emp_id integer primary key, emp_name varchar(20), mgr varchar(20));

INSERT INTO employee values(1001, 'anna', null);
INSERT INTO employee values(1002, 'anthony', 'anna');
INSERT INTO employee values(1003, 'andy', 'sachin');
INSERT INTO employee values(1004, 'sam', 'anna');
INSERT INTO employee values(1005, 'tom', 'sam');
INSERT INTO employee values(1006, 'ricky', 'sam');
INSERT INTO employee values(1007, 'sachin', 'anna');
INSERT INTO employee values(1008, 'amy', 'anthony');
INSERT INTO employee values(1009, 'christina', 'anna');
INSERT INTO employee values(1010, 'jennifer', 'anthony');
```

EMP_ID	EMP_NAME	MGR
1003	andy	sachin
1008	amy	anthony
1004	sam	anna
1007	sachin	anna
1001	anna	-
1002	anthony	anna
1005	tom	sam
1006	ricky	sam
1009	christina	anna
1010	jennifer	anthony

10 rows returned in 0.01 seconds Download

Trigger

```
CREATE OR REPLACE TRIGGER emp_changes
BEFORE UPDATE ON employee
FOR EACH ROW
DECLARE
  CURSOR emp
```

```
  IS
    SELECT * FROM employee WHERE mgr = :OLD.emp_name;
BEGIN
  FOR i IN emp
  LOOP
    UPDATE employee SET mgr = null WHERE mgr = i.emp_id;
  END LOOP;
END;
```

Query

```
DELETE FROM employee WHERE emp_id = 1004;
```

This will delete row successfully
