**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);*

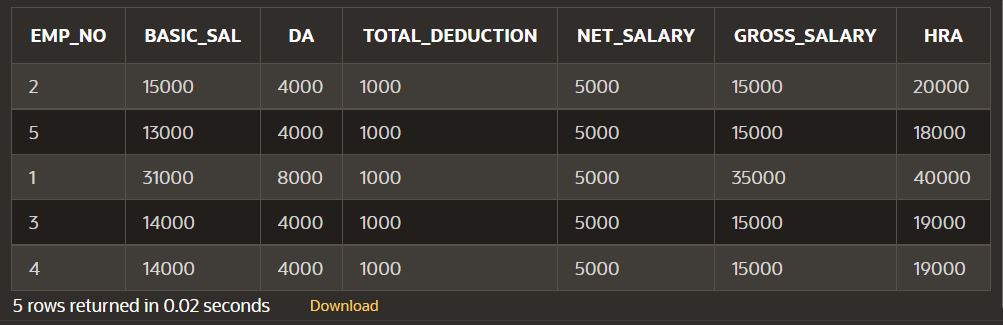
*insert into salary values(2, 15000, 4000, 1000, 5000, 15000, 20000);*

*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);*



**Creating Trigger:**

CREATE OR REPLACE TRIGGER salary\_changes

BEFORE UPDATE ON salary

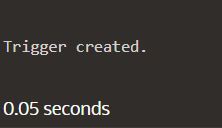
FOR EACH ROW

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);

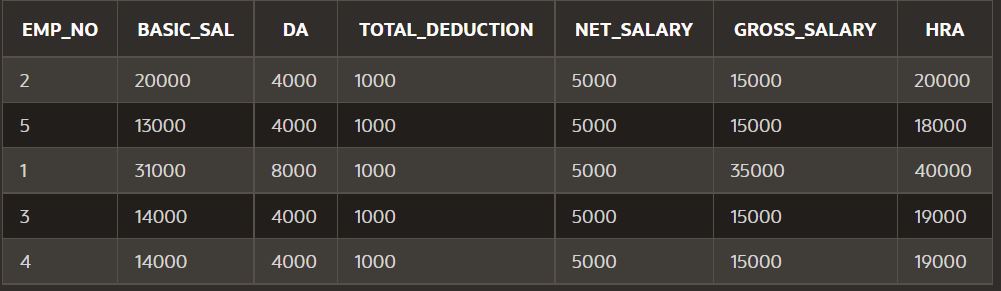
END;



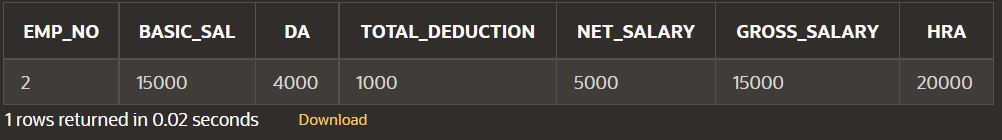
**Quering:**

UPDATE salary SET basic\_sal = 20000 WHERE emp\_no = 2;

**Salary Table after modification:**



**Row inserted after trigger called:**



1. **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);**

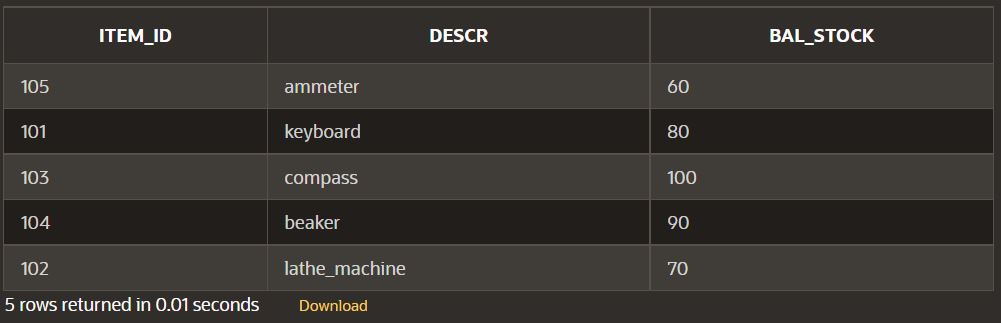
**INSERT into item\_requisite values(102, 'mech', 20);**

**INSERT into item\_requisite values(103, 'civil', 25);**

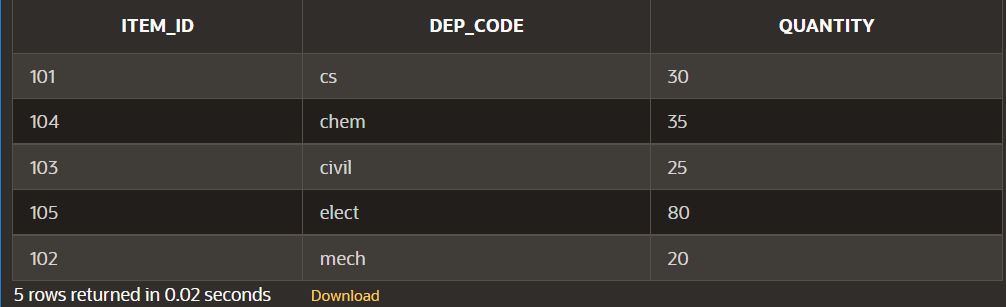
**INSERT into item\_requisite values(104, 'chem', 35);**

**INSERT into item\_requisite values(105, 'elect', 80);**

**Table item\_master :**



**Table item\_requisite:**



**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;

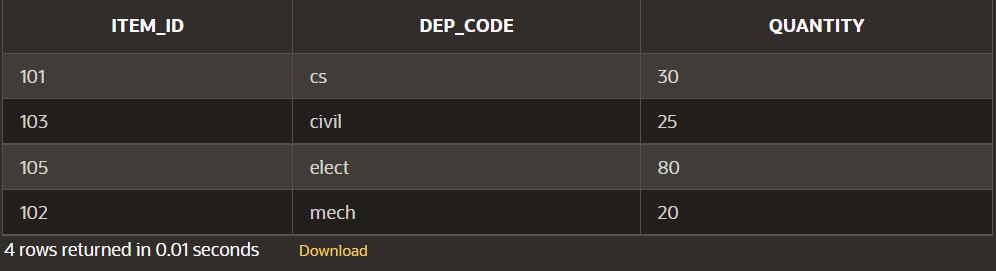
**Query:**

DELETE from item\_requisite where item\_id = 104;

**Table item\_master:**



**Table item\_requisite:**



1. **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');



**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