1. Create a trigger that will show a line on insert operation.

“New Record has been added”

create or replace trigger insert\_emp

after insert on emp

for each row

begin

dbms\_output.put\_line('New Record has been added');

end;

TEST:

insert into emp (empno,ename)

values(7777,'NOUMAN');

New Record has been added

1 row(s) inserted.

0.01 seconds

1. Create a table for log that stores the Date, username doing the operation and operation name (insert, update and delete) on EMP table.

create or replace trigger Log\_OP

before insert or update or delete on emp

for each row

begin

insert into log values (sysdate,user,'operation');

end;

1. Before inserting new record add the commission of employee as 30% of his salary only if the employee is of department no. 10.

create or replace trigger add\_comm

before insert on emp

for each row

BEGIN

if (:new.deptno =10) then

:new.comm :=  :new.sal \* 30/100;

end if;

end;

Test:

insert into emp( empno,ename,job,mgr,hiredate,sal,comm,deptno)

values (7956,'NOmi','clerk',7839,TO\_DATE ('Sep 26 1999','Mon dd yyyy'),200,0,10)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **EMPNO** | **ENAME** | **JOB** | **MGR** | **HIREDATE** | **SAL** | **COMM** | **DEPTNO** |
| 7777 | NOUMAN | - | - | - | - | - | - |
| 7956 | NOmi | clerk | 7839 | 09/26/1999 | 200 | 60 | 10 |

1. Create a trigger that will be fired after updating the salary of an employee. It should display the new sal, old sal and their difference

create or replace trigger sal\_difference

after update on emp

for each row

declare

sal\_diff  number;

begin

sal\_diff  := :new.sal  - :old.sal;

dbms\_output.put\_line('Old salary: ' || :old.sal);

dbms\_output.put\_line('New salary: ' || :new.sal);

dbms\_output.put\_line('Salary difference: ' || sal\_diff);

end;

1. i) Disable the Above inserted trigger

ii)Drop the trigger

1. alter trigger add\_comm disable
2. drop trigger add\_comm
3. Create a new table named EMPSAL that will contain fields for average, min and max salary of emp table and a trigger that will update EMPSAL values on each insertion, updating or deletion in emp table.

I have created separate triggers for each

i)

create  table EMPSAL

(

avg NUMBER(7, 2),

min NUMBER(7, 2),

max NUMBER(7, 2)

)

ii)

insert into empsal

values (0,0,0);

-- maximum

create or replace trigger max\_empsal

after update on emp

for each row

declare

maxim  number;

begin

select max(sal) into maxim from emp;

update empsal

set max =maxim;

end;

--minimum

create or replace trigger min\_empsal

after update on emp

for each row

declare

minim  number;

begin

select min(sal) into minim from emp;

update empsal

set min =minim;

end;

--avg

create or replace trigger avg\_empsal

after update on emp

for each row

declare

averag  number;

begin

select avg(sal) into averag from emp;

update empsal

set min =averag;

end;

1. Using Acc\_master & Acc\_trans table discussed in the lecture ,

Create a trigger on acc\_trans which would be fired before any new transaction. If the transaction is of type ‘W’, you should make sure that a single account holder cannot withdraw more than 20,000 in one day. (If customer withdraws 15,000 once and tries to withdraw 6,000, it should display “you cannot withdraw more than 20,000 in one day”)

i)

create  table acc\_master

(

AccNO NUMBER(4) primary key,

AccBal NUMBER(10,2)

)

ii)

create  table acc\_trans

(

AccNO NUMBER(4) primary key,

transdate NUMBER(10,2),

Amount NUMBER(10,2),

transtype char(1) check(transtype  in('D','W')),

comments varchar2(50)

)

iii)

alter table  acc\_trans

add foreign key(AccNO) references acc\_master(AccNO)

1. Create a trigger on Acc\_trans, If the transaction type is ‘W’ you should make sure that the remaining balance should not be less than 5000.

create  trigger SuccIfNotLessThan

before insert on acc\_trans

for each row

declare

    bal acc\_master.accbal%type;

begin

    select accbal into bal from acc\_master where accno = :new.accno;

    bal := bal - :new.Amount ;

    if (bal<5000) then

        :new.comments ='Transaction fail';

    else

        :new.comments ='Transaction successful';

    end if;

end;

1. Create a trigger on Acc\_trans, If the transaction type is ‘D’,

the transaction should be succeeded only if the amount provided is greater than zero.

create  trigger SuccIfNotZero

before insert on acc\_trans

for each row

declare

    bal acc\_master.accbal%type;

begin

    select accbal into bal from acc\_master where accno = :new.accno;

    if (:new.transtype = 'D') then

        if (:new.amount>0) then

            begin

                update acc\_master

                set accbal = accbal + :new.amount;

                :new.comments := 'Deposit Successful';

            end;

        else

            begin

                := 'Amount Should be greater than Zero';

            end;

        end if;

end if;

end;

1. Display all triggers

select \* from user\_triggers