# **CS 262 – DBMS Lab (A.Y. 2021 – 2022)**

# **Lab Cycle – 4**

1. Write a stored procedure, raise\_salary which accepts an employee number, increment and modifies salary of that employee in employee table. Modified salary = salary increase amount+ current salary. If employee number is not found or if the current salary is null, it should raise an exception. Otherwise, updates the salary.

**PROCEDURE FOR UPDATING SALARY**

create or replace procedure raisesal(n number) as

s number;

a number;

exc exception;

begin

s:=incr(n);

select sal into a from emp2 where empno=n;

if a is null then raise exc;

else update emp2 set sal=sal+s where empno=n;

end if;

exception

when no\_data\_found then

dbms\_output.put\_line('this employee is not exists');

when exc then

dbms\_output.put\_line('salary is null');

end;

**MAIN PROGRAM**

declare

n number:=&no;

r emp2%rowtype;

begin

raisesal(n);

select \* into r from emp2 where empno = n;

dbms\_output.put\_line(r.empno||’ ‘||r.sal);

end;

1. Write a PL/SQL function that accepts department number and returns the total salary of the department.

create or replace function dept\_sal(dno number) return number as

ts number;

begin

select sum(sal) into ts from emp where deptno=dno;

return ts;

end;

1. Write a PL/SQL block that computes increment of an employee in employee table by using incr function which takes employee number as argument, calculates increment and returns the same based on the following criteria:

* If salary <= 3000 – increment = 30% of salary
* If salary > 3000 and <= 6000– increment = 20% of salary
* Else increment = 10% of salary.

**FUNCTION FOR CALCULATING INCREMENT**

create or replace function incr(n number) return number as

s number;

begin

select sal into s from emp2 where empno=n;

if s <= 3000 then s:=0.3\*s;

elsif s>3000 and s<=6000 then

s:=0.2\*s;

else s:=0.1\*s;

end if;

return s;

end;

**MAIN PROGRAM**

declare

n number:=&no;

s number;

begin

s:=incr(n);

dbms\_output.put\_line('incr salary'||s);

update emp2 set sal=sal+s where empno=n;

end;

1. Write a stored procedure that displays the employee names and their total earnings from the Emp Table.*Hint: Total earning of an employee = 12\*(gross\_salary+commission)*

create or replace procedure annual\_pay as

CURSOR ann\_sal is

select ename,12\*(sal+nvl(comm,0)) annual\_sal from emp;

enam varchar2(20);

a\_sal number;

begin

open ann\_sal;

loop

fetch ann\_sal into enam,a\_sal;

dbms\_output.put\_line('Name: '||enam||' Annual sal: '||a\_sal);

exit when ann\_sal%notfound;

end loop;

close ann\_sal;

end;

1. Create a database trigger that checks whether the new salary of employee is less than existing salary. If so, raise an appropriate exception and avoid that updation.

create or replace trigger sal\_check

before update on emp3

for each row

begin

if :new.sal< :old.sal then

raise\_application\_error(-20006,'You Cannot Decrease an emp''s Sal');

end if;

end;

1. Consider the following tables

PERSINFO

|  |  |  |
| --- | --- | --- |
| EMPNO | NAME | AGE |

AUDITPERSINFO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EMPNO | NAME | AGE | OPERATION | ODATE |

PERSINFO is the table for which the auditing must be performed and AUDITPERSINFO is the table which keeps track of the records deleted or modified. Create a database trigger audit\_trial. This trigger is forced when an UPDATE or a DELETE is performed on the table PERSINFO. It first checks for the operation being performed on the table. Then depending on the operation, a variable (that corresponds to operation) is assigned the value ‘UPDATE’ or ‘DELETE’ and then inserts the updated/deleted record into AUDITPERSINFO.

**Program:**

create or replace trigger audit\_trial

after update or delete on persinfo for each row

declare

eno number;

ename varchar2(12);

eage number;

op varchar2(12);

begin

if updating then op:='update';

elsif deleting then op:='delete';

end if;

eno := :old.empno;

ename := :old.name;

eage := :old.age;

insert into auditpersinfo

values(eno,ename,eage,op,sysdate);

end;