Exp-11

Name: Pranaav kumar roll.no:230701233

|  |  |  |
| --- | --- | --- |
| **Ex.No.: 11** | | **PL SQL PROGRAMS** |
| **Date:** | 13/09/2024 |

# PROGRAM 1

Write a PL/SQL block to calculate the incentive of an employee whose ID is 110.

DECLARE

pl\_emp\_id employees.employee\_id%TYPE := 110; pl\_salary employees.salary%TYPE;

pl\_incentive NUMBER; BEGIN

SELECT salary INTO pl\_salary FROM employees

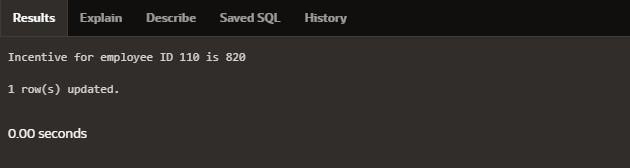
WHERE employee\_id = pl\_emp\_id; pl\_incentive := pl\_salary \* 0.10;

UPDATE employees

SET incentive = pl\_incentive WHERE employee\_id = pl\_emp\_id;

DBMS\_OUTPUT.PUT\_LINE('Incentive for employee ID ' || pl\_emp\_id || ' is ' || pl\_incentive);

COMMIT; END;



# PROGRAM 2

Write a PL/SQL block to show an invalid case-insensitive reference to a quotedand without quoted user-defined identifier.

DECLARE

employeeName VARCHAR2(100); "EmployeeID" NUMBER;

BEGIN

employeeName := 'John Doe'; "EmployeeID" := 40;

DBMS\_OUTPUT.PUT\_LINE('Employee Name: ' || employeeName); DBMS\_OUTPUT.PUT\_LINE('Employee ID: ' || "EmployeeID");

END;



Write a PL/SQL block to adjust the salary of the employee whose ID 122.Sample table: employees

# DECLARE

v\_employee\_id NUMBER := 122; v\_salary NUMBER; v\_new\_salary NUMBER;

v\_increase\_percentage NUMBER := 0.40; BEGIN

SELECT salary INTO v\_salary FROM employees

WHERE employee\_id = v\_employee\_id;

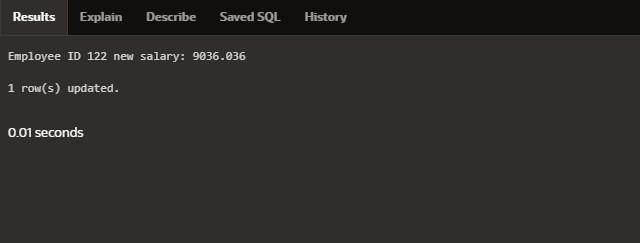
v\_new\_salary := v\_salary + (v\_salary \* v\_increase\_percentage / 100); UPDATE employees

SET salary = v\_new\_salary

WHERE employee\_id = v\_employee\_id;

DBMS\_OUTPUT.PUT\_LINE('Employee ID ' || v\_employee\_id || ' new salary: ' || v\_new\_salary);

# END;



Write a PL/SQL block to create a procedure using the "IS [NOT] NULL Operator" andshow AND operator returns TRUE if and only if both operands are TRUE.

create or replace procedure check\_nullis value1 number := 10;

value2 number := null; begin

if value1 is not null and value2 is null then

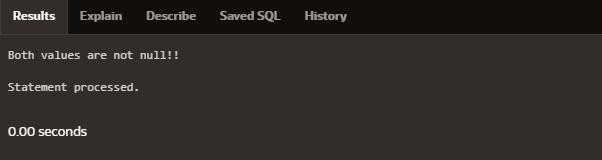
dbms\_output.put\_line('Both values are not null!!'); else

dbms\_output.put\_line('Null value found');end

if; end;

# BEGIN

check\_null; END;



Write a PL/SQL block to describe the usage of LIKE operator including wildcard characters and

escape character.

declare

v\_employeename employees.first\_name%type; v\_employeeid NUMBER := 122;

begin

select first\_name into v\_employeenamefrom employees

where first\_name like '%e%' and employee\_id = v\_employeeid; DBMS\_OUTPUT.PUT\_LINE(v\_employeename);

# END;

Write a PL/SQL program to arrange the number of two variable in such a way that the small number will store in num\_small variable and large number will store in num\_largevariable.

declare

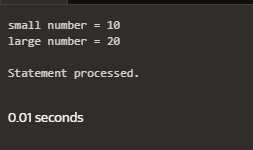
ab number :=10;

cd number :=20; num\_small number; num\_large number; begin

if ab>cd then num\_small :=cd; num\_large :=ab; else

num\_small :=ab; num\_large :=cd; end if;

dbms\_output.put\_line('small number = '||num\_small); dbms\_output.put\_line('large number = '||num\_large); End;



Write a PL/SQL procedure to calculate the incentive on a target achieved and display themessage either the record updated or not.

create or replace procedure calculate\_incentive(p\_emp\_id employees.employee\_id%type, p\_target number)

is

v\_incentive number(7,2); v\_salary employees.salary%type;

begin

select salary into v\_salary from employees

where employee\_id = p\_emp\_id;

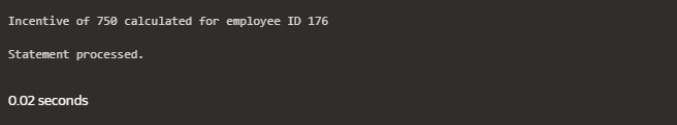
if p\_target >= 100000 then v\_incentive := v\_salary \* 0.1;

dbms\_output.put\_line('Incentive of ' || v\_incentive || ' calculated for employee ID ' || p\_emp\_id);

else

dbms\_output.put\_line('No incentive for employee ID ' || p\_emp\_id);end if;

End;



Write a PL/SQL procedure to calculate incentive achieved according to the specific salelimit.

create or replace procedure incentive\_sale(p\_emp\_id employees.employee\_id%type,p\_sales number)

is

v\_incentive number(7,2); begin

if p\_sales > 100000 then v\_incentive := p\_sales \* 0.1;

elsif p\_sales between 50000 and 100000 then v\_incentive := p\_sales \* 0.05;

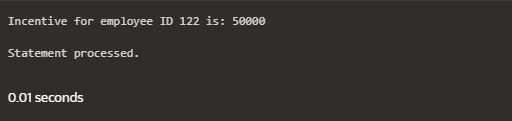
else

v\_incentive := 0; end if;

dbms\_output.put\_line('Incentive for employee ID ' || p\_emp\_id || ' is: ' || v\_incentive);End;

begin

incentive\_sale(122,500000); end;



Write a PL/SQL program to count number of employees in department 50 and checkwhether this department have any vacancies or not. There are 45 vacancies in this department.

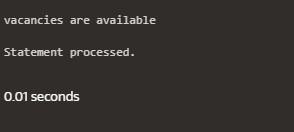
declare

no\_of\_emp number; vacancies number:=45; begin

select count(\*) into no\_of\_emp from employees where department\_id=50;if no\_of\_emp<vacancies then

dbms\_output.put\_line('vacancies are available');else dbms\_output.put\_line('vacancies are not available');end if;

end;



# PROGRAM 10

Write a PL/SQL program to count number of employees in a specific department and check whether this department have any vacancies or not. If any vacancies, how manyvacancies are in that department.

declare

v\_department\_id number := 55; v\_emp\_count number; v\_vacancies number := 50;

begin

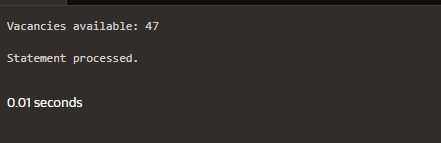
select count(\*) into v\_emp\_count from employees

where department\_id = v\_department\_id;

if v\_emp\_count < v\_vacancies then

dbms\_output.put\_line('Vacancies available: ' || (v\_vacancies - v\_emp\_count));else dbms\_output.put\_line('No vacancies available.');end

if; end;



Write a PL/SQL program to display the employee IDs, names, job titles, hire dates, andsalaries of all employees.

begin

for i in (select employee\_id, first\_name || ' ' || last\_name as name, job\_id, hire\_date,salary from employees)

loop

dbms\_output.put\_line('ID: ' || i.employee\_id || ', Name: ' || i.name || ', Job: ' || i.job\_id

|| ', Hire Date: ' || i.hire\_date || ', Salary: ' || i.salary);end loop;

end;



Write a PL/SQL program to display the employee IDs, names, and department names ofall employees.

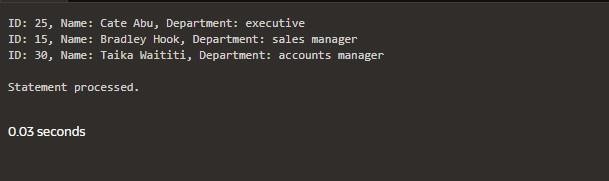
begin

for i in (select e.employee\_id, e.first\_name || ' ' || e.last\_name as name, d.dept\_namefrom employees e

join department d on e.employee\_id = d.dept\_id) loop dbms\_output.put\_line('ID: ' || i.employee\_id || ', Name: ' || i.name || ', Department: ' ||

i.dept\_name); end loop;

End;



Write a PL/SQL program to display the job IDs, titles, and minimum salaries of all jobs.

begin

for rec in (select e.employee\_id, d.dept\_name, min(salary) as min\_salary from employees

e join department d

on e.employee\_ID = d.dept\_id

group by e.employee\_id , d.dept\_name)loop

dbms\_output.put\_line('Job ID: ' || rec.employee\_id || ', Title: ' || rec.dept\_name || ',Min Salary: ' || rec.min\_salary);

end loop; End;



begin

for rec in (select e.employee\_id, d.dept\_name, min(salary) as min\_salary from employees

e join department d

on e.employee\_ID = d.dept\_id

group by e.employee\_id , d.dept\_name)loop

dbms\_output.put\_line('Job ID: ' || rec.employee\_id || ', Title: ' || rec.dept\_name || ',Min Salary: ' || rec.min\_salary);

end loop; End;



# PROGRAM 14



Write a PL/SQL program to display the employee IDs, names, and job history start datesof all Employees.

Begin

for rec in (select employee\_id, first\_name || ' ' || last\_name as name, hire\_datefrom employees) loop

dbms\_output.put\_line('ID: ' || rec.employee\_id || ', Name: ' || rec.name || ', Start Date: '

|| rec.hire\_date); end loop;

end;

# PROGRAM 15

**BEGIN**

FOR rec IN (SELECT employee\_id, first\_name || ' ' || last\_name AS name, end\_dateFROM employees)

# LOOP

dbms\_output.put\_line('ID: ' || rec.employee\_id ||',

Name: ' || rec.name || ', End Date: ' ||

NVL(TO\_CHAR(rec.end\_date, 'YYYY-MM-DD'), 'Still Active'));

**END LOOP; END;**

