

## PROGRAM 1

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

Declare :

V-Salary employees, salary % type;  
V-Incentive number;

Begin

```
Select salary into V-Salary from  
employees where employee_id = 110;  
If V-Salary > 10000 then  
    V-Incentive := V-Salary * 0.10  
Else If V-Salary between 5000 and 10000 then  
    V-Incentive := V-Salary * 0.07;  
Else  
    V-Incentive := V-Salary * 0.05;  
End If;  
DBMS-output.put_line ("Incentive : ||V-Incentive);  
End;
```

1

## PROGRAM 2

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

```
Declare
    "Name" Varchar2(20) := 'Alice';
Begin
    DBMS_OUTPUT.PUT_LINE ('Name');
End;
/
```

### PROGRAM 3

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

Sample table: employees

Declare

v\_salary employees.salary %TYPE;

Begin

select salary into v\_salary from  
employees where employee\_id = 122;

v\_salary := v\_salary \* 1.10;

update employees

set salary = v\_salary

where employee\_id = 122;

DBMS.output.put\_line ("Salary updated");

commit;

End ;

## PROGRAM 4

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

## PROGRAM 5

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

Declare

V\_name Varchar2(20) := 'A-Raju';

Begin

IF V\_name like 'A . .' then

DBMS\_output.put\_line ('Name starts with A');

End if,

If V\_name like 'A ..' then

DBMS\_output.put\_line ('Name starts  
with A atleast two');

End if;

End;

/

## PROGRAM 6

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\_large variable.

Declare

num1 Number := 25;

num2 Number := 10;

num\_small number;

num\_large number;

BEGIN

IF num1 < num2 THEN

    num\_small := num1;

    num\_large := num2;

ELSE

    num\_small := num2;

    num\_large := num1;

END IF,

END;

/

## PROGRAM 7

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

```
set serveroutput on;
/
create or replace procedure calc
  p_empid IN employees.emploideid%type
  p_target IN number
) IS
  v_incentive number;
  v_count number;
Begin
  if p_target >= 100 then
    v_incentive := 1000;
  else
    v_incentive := 1500
  end if;
end;
/
```

## PROGRAM 8

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

```
set serveroutput on;
create or replace procedure calc (
    p_sales IN Number)
IS
    v_incentive Number := 0;
Begin
    if p_sales >= 10000 then
        v_incentive := 1000;
    else if p_sales >= 75000 then
        v_incentive := 7500;
    else if p_sales >= 50000 then
        v_incentive := 4000;
    else
        v_incentive := 1000;
    end if;
End;
/
```

## PROGRAM 9

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

Declare

V-emp-count Number;

V-Vacancies Number := 45;

Begin

Select count(\*) into V-emp-count  
from employees  
where department\_id = 50;

DBMS\_OUTPUT.PUT\_LINE ('Employees');

If V-emp-count < V-Vacancies then

DBMS\_OUTPUT.PUT\_LINE ('Available');

Else

DBMS\_OUTPUT.PUT\_LINE ('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 many vacancies are in that department.

Declare

V\_dept\_id Number := 15;

V\_emp\_count Number;

V\_total\_positions number := 45;

V\_Vacancies number;

Begin :

Select count (\*) INTO V\_emp\_count

from employees;

V\_Vacancies := V\_total\_position - V\_emp\_count;

If V\_Vacancies then

else

DBMS\_output.put\_line ('Not available');

end ;

/

## PROGRAM 11

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

```
Set serveroutput on;
/
BEGIN
    for emp IN (select employee_id, first_name||'
    last_name' AS emp_name, job_id, hire_date,
    from employee) LOOP
        DBMS_output.put_line (emp);
    end loop;
END;
/
```

## PROGRAM 12

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

```
SET SERVEROUTPUT ON;
/
BEGIN
  FOR emp IN (SELECT e.employee_id, c.first
 -name || ' ' || c.last_name AS emp_name, d.
department
  FROM employees) LOOP
    DBMS_OUTPUT.PUT_LINE (emp);
  END LOOP;
END;
/
```

## PROGRAM 13

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

```
SET SERVEROUTPUT ON;
/
BEGIN
    for Job_rec IN (select job_id, job_title, min-
                     salary from jobs) loop
        DBMS_OUTPUT.PUT_LINE (Job_rec);
    end loop;
END;
/
```

## PROGRAM 14

Write a PL/SQL program to display the employee IDs, names, and job history start dates of all employees.

```
SET SERVEROUTPUT ON;
/
BEGIN
    FOR emp_list IN (SELECT e.employee_id
                      e.first_name, e.last_name, e.start
                      FROM employees e) LOOP
        DBMS_OUTPUT.PUT_LINE (emp_list);
    END LOOP;
END;
/
```

## PROGRAM 15

Write a PL/SQL program to display the employee IDs, names, and job history end dates of all employees.

```
SET SERVER OUTP DT ON;
/
BEGIN
    FOR emp_rec IN (SELECT e.employee_id,
                           e.first_name, c.last_name, e.end_dt_job
                          FROM employee e)
    DBMS_OUTPUT.PUT_LINE (emp_rec);
END LOOP;
END;
```

Evaluation Procedure	Marks awarded
PL/SQL Procedure(5)	5
Program/Execution (5)	5
Viva(5)	5
Total (15)	15
Faculty Signature	Ran

### Program 1

#### FACTORIAL OF A NUMBER USING FUNCTION

Set screenoutput on;  
Create or replace function factorial (n number)  
return number  
is

fact number := 1;

Begin

for i in 1..n loop

fact := fact \* i;

End loop

return fact;

End;

/

Declare

num Number := 5;

result Number := 1;

Begin

result := factorial (num);

DEMSI-output.putLine ('factorial of ' || num || ' is ' || result);

End;

## Program 2

Write a PL/SQL program using Procedure IN,INOUT,OUT parameters to retrieve the corresponding book information in library.

Get user input  
Create or replace procedure get\_book\_info(book\_id in No,  
, P\_bookname out VARCHAR2, PAuthor out VARCHAR2, PPrice  
out NUMBER) is begin  
SELECT book\_name, author, price into pbookid, PAuthor,  
PPrice FROM Library WHERE book\_id = pbook\_id;

PL/SQL - cursor, OUT clause (Book\_name || &book\_name),  
DECODE - cursor FOR loop (Author || &author),  
DBMS\_OUTPUT.PUT\_LINE twice || price || price)

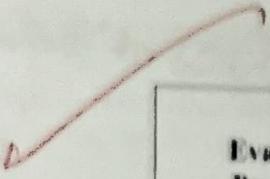
exception

when no\_data\_found then

DBMS\_OUTPUT.PUT\_LINE('No book found with Id');

Procedure 2:

```
END;
DECLARE
    V_book_name VARCHAR2(20);
    V_author VARCHAR2(20),
    V_price NUMBER(2,2)
```



Evaluation Procedure	Marks awarded
PL/SQL Procedure(5)	5
Program/Execution (5)	5
Viva(5)	5
Total (15)	15
Faculty Signature	Rajesh