PROGRAM 1

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

```
Declare
   id    number  := 110
   sal   number ;
   inc   number ;
Begin Select into sal
      from employees
      where emp_id = id
      inc := sal * 0.10;
      DBMS. output. put_line ( ' Incentative = ' || inc);
Exceptions
      when NO_DATA_FOUND Then
            DBMS_output.put_Line ('No such employee')
End
```

PROGRAM 2

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

```
Declare
    "xvar" number := 10;
Begin
    dbms.output.put_line (xvar);
End;
/
```

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

Sample table: employees

```
Declare
    id number := 122;
    sal number;
Begin
    Select salary into sal
    from employees
    where employee_id = id;
    sal := sal + (sal * 0.10);
    update employees
    Set salary = sal
    where employee_id = id;
    DBMS_OUTPUT.PUT_LINE ('Salary updated to ' || sal);
Exception
        When no_data_found then
            DBMS_OUTPUT.PUT_LINE ('No such employee');
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.

```
Create or replace procedure chk_null (a
in number, b in number) Is
Begin
If a is not null & b is not null then
dbms_output.put_line('True')
Else
dbms_output.put_line('False');
END;
/
```

PROGRAM 5

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

```
Declare
    n varchar2(20) := 'A_B';
Begin
    If n like 'A%' then
        dbms.output.put_line('% match');
    End if;
    If n Like 'A_B' then
        dbms_output.put_line('_match');
    End if;
    If n like 'A\_B' Escape '\' then
        dbms_output.put_line('escape match')
    end if;
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_large variable.

```
Declare
    a number := 30;
    b number := 10;
    num_small NUMBER;
    num_large NUMBER;
Begin
    If a < b then
        num_small := a;
        num_large := b;
    else
        num_small := b;
        num_large := a;
    End if
    dbms.output.put_line ('small =' || num_small);
    dbms.output.put_line ('large =' || num_large);
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.

```
Create procedure inc_calc (id IN Number, tgt in number) is
        sal number;
        inc Number;
Begin
        select salary into salary
        from employees.
        where employee_id = id
If tgt >= 100 then
        inc := sal * 0.10;
END IF;
        update employees
        set salary = salary + inc
        where employee_id = id;
END
```

PROGRAM 8

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

```
Create procedure inc_sale (sale in number, lim in
number) is
Begin
    if sale >= lim then
        inc := sale * 0.10;
    End if;
    dbms_output.put_line('Incentive = '|| inc);
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
    Cnt  number
    Vac  number := 45
Begin
    select Conunt (*) into cnt
    from employees
    where dept_id = 50
if cnt< Vac then
    dbms_output.put_line('Valice available');
Else
    dbbms_output.put_line('No vacancies');
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
    d number := 50
    tot number := 50
    Cnt number;
    Vac number;
Begin
    select Count(*) into Cnt
    from employees
    where dept_id = d
    Vac := tot - client

End IF;
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;

Declare

    r emp%RowType

Begin

    for r in (select empno, ename, job, hiredate,
    sal from emp) Loop DBMS_OUTPUT.PUT_LINE(

            'ID:' || r.empno ||

            ' | Name:' || r.ename ||

            ' | Job:' || r.job ||

        ' | Hire Date:' || r.hiredate ||

        ' | Salary:' || r.sal );

    end loop;

END;
```

RAM 12

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

```
Set serveroutput ON;
begin
    for r in (
        select e.emp_id, e.first_name,
d.department_name from employees e join
departments d on e.dept_id =
d.dept_id
    )
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 r in (select job_id, job_title,
    min_salary from jobs) Loop
        DBMS_output.put_line (r.job_id
|| '-' || r.job_title || '-' || r.min_sal);
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 SERVERLET ON;
Begin
    select e.emp-id;
    e.firstname || ' ' || e.lastname as name, j.startdate
from employees e join job_history j on e.emp-id
= j.employee_id ) Loop dbms_output.put_line (
        'ID:' || r.employee_id || 'Name:' ||
        r.name || ' | job history start: || r.st
        _date);
    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 serveroutput ON;
begin
    for x in (select emp_id, end_date from
        Job_history ) Loop
dbms.output.put_line (x.employee_id || x.end_
 date);
    end loop;
end;
```

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