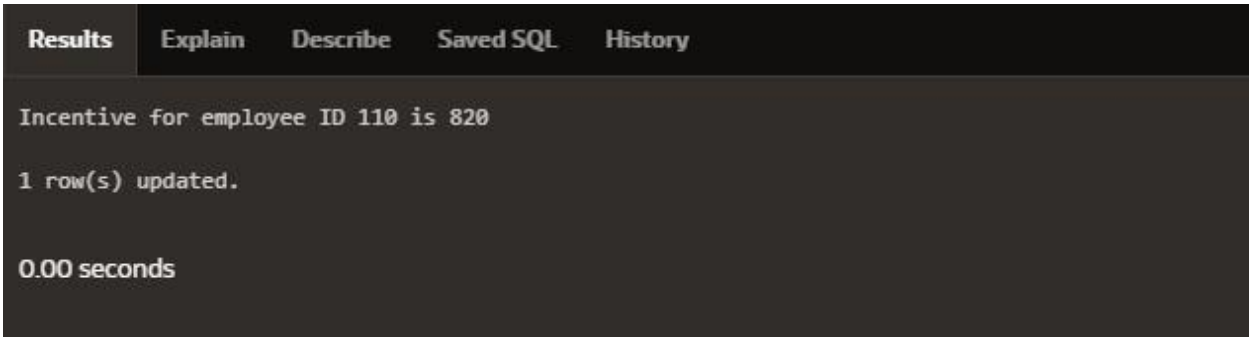
Ex.No.: 11		PL SQL PROGRAMS
Date:	13/09/2024	

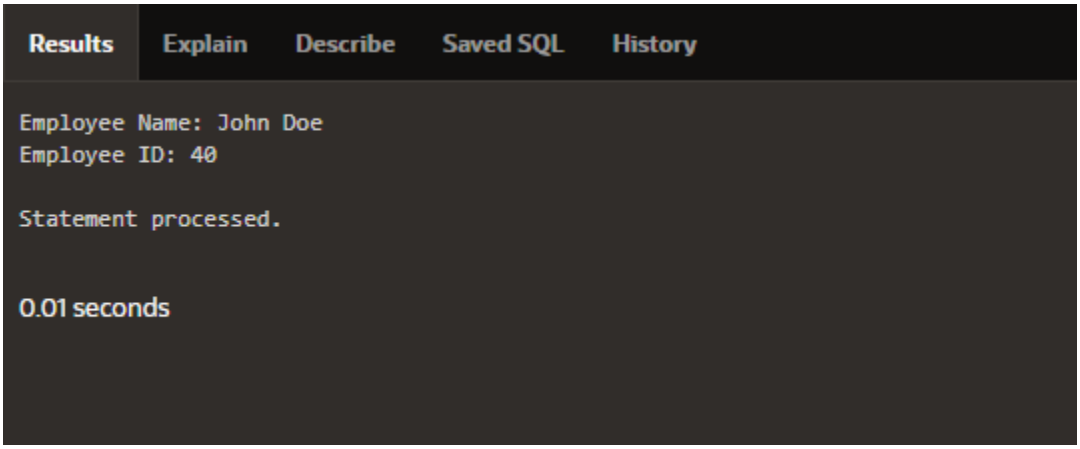
PROGRAM 1
Write a PL/SQL block to calculate the incentive of an employee whose ID is 110.

```
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 quoted and without quoted user-defined identifier.

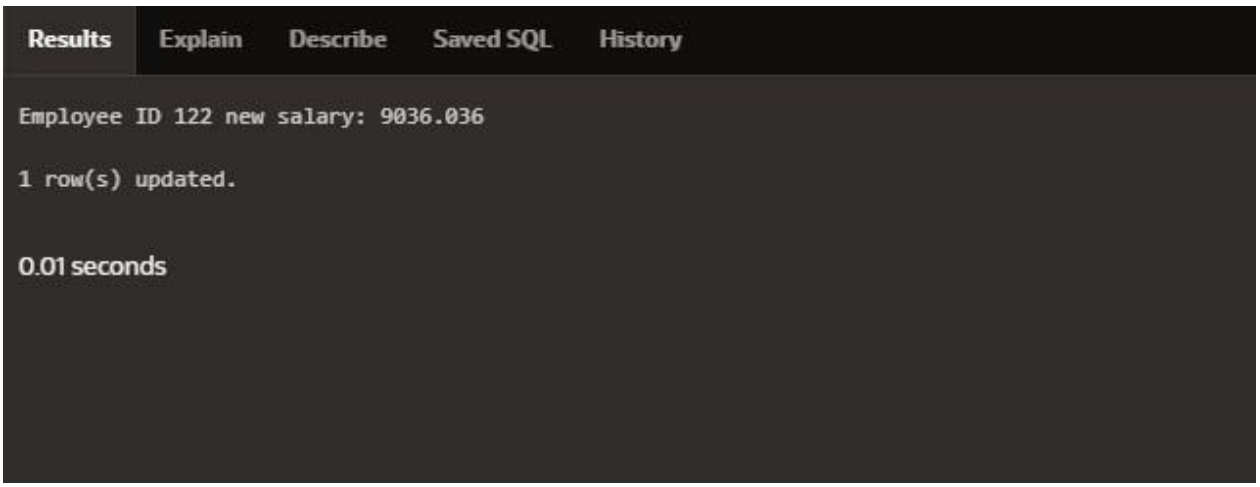
```
DECLARE
  employeeName VARCHAR2(100);
  "EmployeeID"  NUMBER;
BEGIN
  employeeName := 'John Doe';
  "EmployeeID" := 40;

  DBMS_OUTPUT.PUT_LINE('Employee Name: ' || employeeName);
  DBMS_OUTPUT.PUT_LINE('Employee ID: ' || "EmployeeID");
END;
```



PROGRAM 3

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;
```

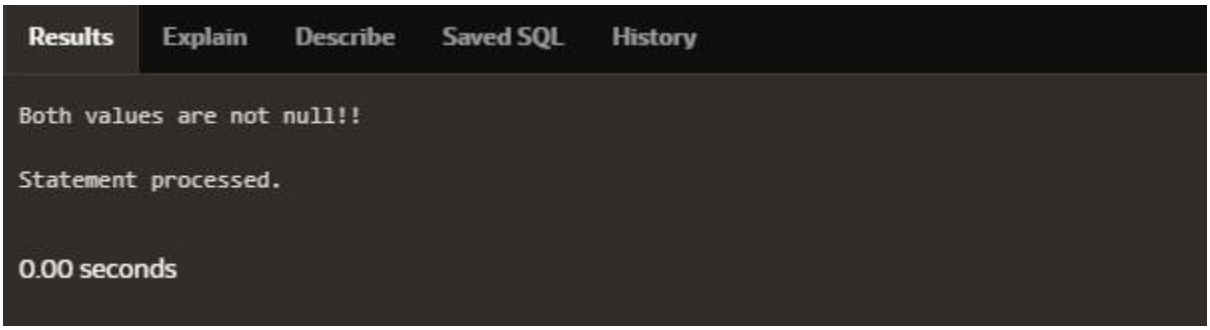


PROGRAM 4

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 check_null
is
   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;
```



PROGRAM 5

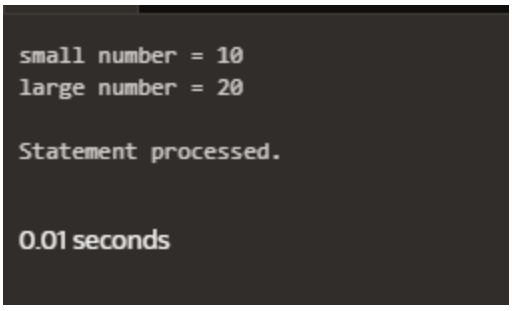
**Write a PL/SQL block to describe the usage of LIKE operator including wildcard characters and
escape character.**

```
declare

v_employeename employees.first_name%type;
v_employeeid NUMBER := 122;

begin

select first_name into v_employeename
from employees
where first_name like '%e%' and employee_id = v_employeeid;

DBMS_OUTPUT.PUT_LINE(v_employeename);

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
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;
```



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 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;
```

```
Incentive of 750 calculated for employee ID 176

Statement processed.

0.02 seconds
```

PROGRAM 8

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

```
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;
```

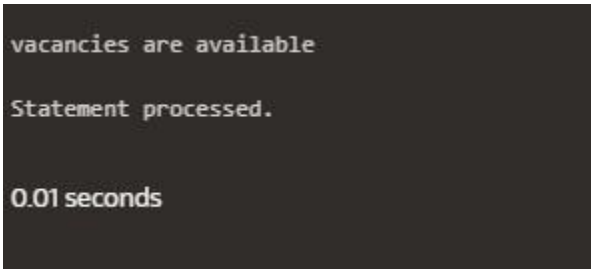
```
begin
   incentive_sale(122,500000);
end;
```

```
Incentive for employee ID 122 is: 50000

Statement processed.


0.01 seconds
```

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.

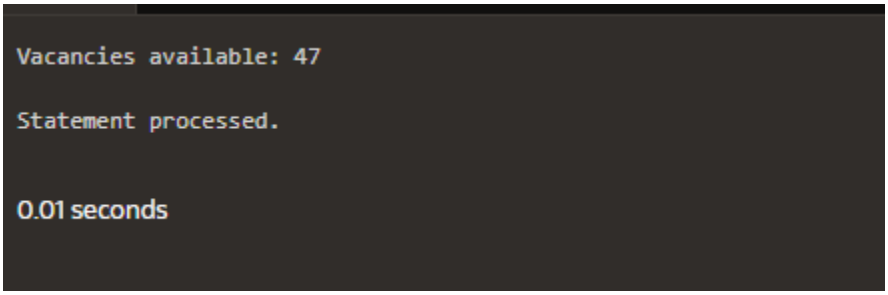
```
declare
no_of_emp number;
vacancies number:=45;
begin
select count(*) into no_of_emp from employees where department_id=50;
if no_of_emp<vacancies then
dbms_output.put_line('vacancies are available');
else
dbms_output.put_line('vacancies are not available');
end if;
end;
```



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 many vacancies 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;
```



PROGRAM 11

Write a PL/SQL program to display the employee IDs, names, job titles, hire dates, and salaries 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;
```

```
ID: 2, Name: Emma Austen, Job: ST_CLERK, Hire Date: 11/06/1990, Salary: 5500
ID: 10, Name: Paul Rudd, Job: #pr010, Hire Date: 04/06/1969, Salary: 2500
ID: 11, Name: Brie Zlotkey, Job: #bl011, Hire Date: 10/01/1989, Salary: 7200
ID: 20, Name: Elizabeth Olsen, Job: #eo020, Hire Date: 02/16/1989, Salary: 7300
ID: 25, Name: Cate Abu, Job: #cb025, Hire Date: 05/14/1969, Salary: 13500
ID: 27, Name: Jeff Goldblum, Job: ST_CLERK, Hire Date: 10/22/1952, Salary: 3500
ID: 122, Name: Robert Downey, Job: #rd003, Hire Date: 04/04/1965, Salary: 9036.04
ID: 18, Name: Karen Gillan, Job: #kg018, Hire Date: 11/28/1987, Salary: 6900
ID: 21, Name: Anthony Mackie, Job: ST_CLERK, Hire Date: 09/23/1978, Salary: 4000
ID: 22, Name: Sebastian Stan, Job: #ss022, Hire Date: 08/13/1982, Salary: 9000
ID: 28, Name: Karl Austin, Job: #ka028, Hire Date: 06/07/1972, Salary: 13500
ID: 176, Name: Chris Morris, Job: #ce005, Hire Date: 05/07/1994, Salary: 7500
ID: 6, Name: Mark Ruffalo, Job: #mr006, Hire Date: 11/22/1967, Salary: 7200
ID: 12, Name: Chadwick Boseman, Job: #cb012, Hire Date: 11/29/1976, Salary: 8000
ID: 24, Name: Tom Hiddleston, Job: #th024, Hire Date: 02/09/1981, Salary: 6500
ID: 1, Name: Justin Beiber, Job: ST_CLERK, Hire Date: 09/21/1996, Salary: 4900
ID: 8, Name: Jeremy Wilson, Job: #ja008, Hire Date: 01/07/1971, Salary: 13500
ID: 7, Name: Chris Hemsworth, Job: #ch007, Hire Date: 08/11/1983, Salary: 7800
ID: 9, Name: Tom Holland, Job: ST_CLERK, Hire Date: 06/01/1996, Salary: 6000
ID: 13, Name: Chris Austin, Job: #ca013, Hire Date: 06/21/1979, Salary: 13500
ID: 17, Name: Dave Bautista, Job: #db017, Hire Date: 01/18/1969, Salary: 6500
ID: 26, Name: Tessa Thompson, Job: ST_CLERK, Hire Date: 10/03/1983, Salary: 5200
ID: 14, Name: Zoe Austin, Job: #za014, Hire Date: 06/19/1978, Salary: 13500
ID: 19, Name: Pom Davies, Job: #pk019, Hire Date: 05/03/1986, Salary: 1100
ID: 42, Name: Matos roy, Job: #mr042, Hire Date: 02/23/1991, Salary: 7000
ID: 4, Name: Scarlett Austin, Job: #sa004, Hire Date: 11/22/1984, Salary: 13500
ID: 15, Name: Bradley Hook, Job: ST_CLERK, Hire Date: 01/05/1975, Salary: 4500
ID: 16, Name: Vin Diesel, Job: #vd016, Hire Date: 07/18/1967, Salary: 8000
ID: 110, Name: Benedict andru, Job: #bc023, Hire Date: 07/19/1976, Salary: 8200
ID: 30, Name: Taika Waititi, Job: #tw030, Hire Date: 08/16/1975, Salary: 7700
ID: 40, Name: John Doe , Job:   #jd040 , Hire Date: 08/10/1995, Salary: 6000
ID: 29, Name: Idris Elba, Job: #ie029, Hire Date: 09/06/1972, Salary: 7400
ID: 41, Name: Matos charles, Job: #mc041, Hire Date: 09/18/1993, Salary: 8900

Statement processed.
```

PROGRAM 12

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

```
begin
   for i in (select e.employee_id, e.first_name || ' ' || e.last_name as name, d.dept_name
            from 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;
```

```
ID: 25, Name: Cate Abu, Department: executive
ID: 15, Name: Bradley Hook, Department: sales manager
ID: 30, Name: Taika Waititi, Department: accounts manager

Statement processed.


0.03 seconds
```

PROGRAM 13

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;
```

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;
```

PROGRAM 14

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

```
Begin
   for rec in (select employee_id, first_name || ' ' || last_name as name, hire_date
               from employees) loop
      dbms_output.put_line('ID: ' || rec.employee_id || ', Name: ' || rec.name || ', Start Date: '
|| rec.hire_date);
   end loop;
end;
```

```
ID: 2, Name: Emma Austen, Start Date: 11/06/1990
ID: 10, Name: Paul Rudd, Start Date: 04/06/1969
ID: 11, Name: Brie Zlotkey, Start Date: 10/01/1989
ID: 20, Name: Elizabeth Olsen, Start Date: 02/16/1989
ID: 25, Name: Cate Abu, Start Date: 05/14/1969
ID: 27, Name: Jeff Goldblum, Start Date: 10/22/1952
ID: 122, Name: Robert Downey, Start Date: 04/04/1965
ID: 18, Name: Karen Gillan, Start Date: 11/28/1987
ID: 21, Name: Anthony Mackie, Start Date: 09/23/1978
ID: 22, Name: Sebastian Stan, Start Date: 08/13/1982
ID: 28, Name: Karl Austin, Start Date: 06/07/1972
ID: 176, Name: Chris Morris, Start Date: 05/07/1994
ID: 6, Name: Mark Ruffalo, Start Date: 11/22/1967
ID: 12, Name: Chadwick Boseman, Start Date: 11/29/1976
ID: 24, Name: Tom Hiddleston, Start Date: 02/09/1981
ID: 1, Name: Justin Beiber, Start Date: 09/21/1996
ID: 8, Name: Jeremy Wilson, Start Date: 01/07/1971
ID: 7, Name: Chris Hemsworth, Start Date: 08/11/1983
ID: 9, Name: Tom Holland, Start Date: 06/01/1996
ID: 13, Name: Chris Austin, Start Date: 06/21/1979
ID: 17, Name: Dave Bautista, Start Date: 01/18/1969
ID: 26, Name: Tessa Thompson, Start Date: 10/03/1983
ID: 14, Name: Zoe Austin, Start Date: 06/19/1978
ID: 19, Name: Pom Davies, Start Date: 05/03/1986
ID: 42, Name: Matos roy, Start Date: 02/23/1991
ID: 4, Name: Scarlett Austin, Start Date: 11/22/1984
ID: 15, Name: Bradley Hook, Start Date: 01/05/1975
ID: 16, Name: Vin Diesel, Start Date: 07/18/1967
ID: 110, Name: Benedict andru, Start Date: 07/19/1976
ID: 30, Name: Taika Waititi, Start Date: 08/16/1975
ID: 40, Name: John Doe , Start Date: 08/10/1995
ID: 29, Name: Idris Elba, Start Date: 09/06/1972
```

PROGRAM 15
Write a PL/SQL program to display the employee IDs, names, and job history end datesof all employees.

```
BEGIN
   FOR rec IN (SELECT employee_id, first_name || ' ' || last_name AS name, end_date
           FROM 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;
```