

Ex.No.: 11		PL SQL PROGRAMS
Date:	09.10.2024	

PROGRAM 1

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

```

declare
a employees.salary%type;
b employees.salary%type;
begin
Select salary into a from employees where employee_id = 110;
b:=0.05*a;
dbms_output.put_line('Salary after incentive : '||(a+b));
end;

```

```
Salary after incentive : 6300
```

```
Statement processed.
```

```
0.01 seconds
```

PROGRAM 2

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

```
declare
non_quoted_variable varchar2(10) := 'Hi';
"quoted_variable" varchar2(10) := 'Hello';
begin
dbms_output.put_line(NON_QUOTED_VARIABLE);
dbms_output.put_line("quoted_variable");
dbms_output.put_line("QUOTED_VARIABLE");
end;
```

```
Hi
Hello
```

```
Statement processed.
```

```
ORA-06550: line 7, column 23:
PLS-00201: identifier 'QUOTED_VARIABLE' must be declared
ORA-06550: line 7, column 1:
PL/SQL: Statement ignored
```

PROGRAM 3

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

```
declare
old_salary employees.salary%type;
new_salary employees.salary%type;
begin
new_salary:= :sal;
Select salary into old_salary from employees where employee_id = 122;
dbms_output.put_line('Before updation: '||old_salary);
Update employees set salary = salary + new_salary where employee_id = 122;
Select salary into new_salary from employees where employee_id = 122;
dbms_output.put_line('After updation: '||new_salary);
end;
```

Before updation: 8000

After updation: 9000

Statement processed.

0.00 seconds

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 proc1( a boolean, b boolean) IS
BEGIN
if(a is not null) and (b is not null) then
if(a = TRUE and b = TRUE) then
dbms_output.put_line('TRUE');
else
dbms_output.put_line('FALSE');
end if;
else
dbms_output.put_line('NULL VALUES in arguments');
end if;
end proc1;

BEGIN
proc1(TRUE,TRUE);
proc1(TRUE,FALSE);
proc1(NULL,NULL);
end;
```

```
TRUE
FALSE
NULL VALUES in arguments
```

```
Statement processed.
```

```
0.00 seconds
```

PROGRAM 5

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

```
Declare
name varchar2(20);
num number(3);
Begin
num := :n;
Select first_name into name from employees where employee_id=num;
if name like 'D%' then
dbms_output.put_line('Name starts with "D"');
end if;
if name like 'Dan_e1%' then
dbms_output.put_line('Name contains "Dan" followed by one character');
end if;
name := 'Daniel_Andrea';
if name like 'Daniel\ _Andrea' escape '\' then
dbms_output.put_line('Name contains "Daniel_Andrea"');
end if;
end;
```

Name starts with "D"

Name contains "Dan" followed by one character

Name contains "Daniel_Andrea"

Statement processed.