

## AIM:

To execute the PL/SQL programs based on the given constraints

5. Develop the PL/SQL programs based on the given constraints.
  - (a) Create a cursor, which displays list of products bought by a customer having cust\_id as 5.
  - (b) Create a cursor, which displays names of employees whose basic salary is between 10,000 and 20,000.
6. Develop the PL/SQL programs to demonstrate the concept of procedure.
  - (a) Create a procedure to update the salaries of all employees 20% in their basic pay.
  - (b) Create a procedure to demonstrate IN, OUT and INOUT parameters.
7. Develop the PL/SQL programs to demonstrate the concept of functions.

70

NATIONAL ENGINEERING COLLEGE, KOVILPATTI  
*An Autonomous Institution, Affiliated to Anna University, Chennai*

- (a) Create a function to find sum of salaries of all employees working in department id 10.
8. Construct a PL/SQL program to retrieve customer name whose customer\_id as 101 using %type and %rowtype attributes.

### 5) a)

```
1  Declare
2  c_id customers.cust_id%type;
3  c_name customers.customer_name%type;
4  p_name products.product_name%type;
5  cursor customerdetails is
6  select c.cust_id, c.customer_name, p.product_name from customers c inner join products p on p.cust_id= c.cust_id where c.cust_id=5;
7  Begin
8  Open customerdetails;
9  Loop
10 Fetch customerdetails into c_id, c_name, p_name;
11 exit when customerdetails%notfound;
12 dbms_output.put_line(c_id || ' ' || c_name || ' ' || p_name);
13 End loop;
14 Close customerdetails;
15 End;
```

Results Explain Describe Saved SQL History

5 John Doe Product A  
5 Alice Johnson Product A  
5 Mary Brown Product A

Activate Windows  
Go to Settings to activate Windows.

5) b)

```
1  Declare
2  Cursor employee_cursor Is
3  Select employee_name from employees
4  where basic_salary between 1000 and 20000;
5  Emp_name employees.employee_name%type;
6  Begin
7  Open employee_cursor;
8  loop
9  Fetch employee_cursor into Emp_name;
10 exit when employee_cursor%notfound;
11 dbms_output.put_line('Employee Name: ' || Emp_name);
12 End Loop;
13 Close employee_cursor;
14 End;
```

Results Explain Describe Saved SQL History

Employee Name: Jane Doe  
Employee Name: Alice Johnson  
Employee Name: John Smith

6) a)

```
1  CREATE OR REPLACE PROCEDURE update_salaries IS
2  BEGIN
3      UPDATE employees
4      SET basic_salary = basic_salary * 1.20;
5  END;
6  /
7
8
```

6) b)

```
1  declare
2  x number;
3  y number;
4  z number;
5  procedure findmin(a IN number, b IN number, c OUT number) is
6  begin
7  if a < b then
8  c := a;
9  else
10 c := b;
11 end if;
12 end;
13 begin
14 x := 10;
15 y := 20;
16 findmin(x, y, z);
17 dbms_output.put_line('MINIMUM NUMBER IS: ' || z);
18 end;
```

Results Explain Describe Saved SQL History

MINIMUM NUMBER IS: 10  
Statement processed.

7) a)

```

Create or REPLACE Function sum_of_salaries
return number is total_salary number:=0;
Begin
Select Sum(basic_salary) into total_salary From employees where employee_id = 101;
return total_salary;
End sum_of_salaries;
DECLARE
sum_salary Number;
Begin
sum_salary:=sum_of_salaries;
dbms_output.put_line('Total salary of employee Department Id 101: ' || sum_salary);
End;
```

```

Results Explain Describe Saved SQL History

Total salary of employee Department Id 101: 17000

Statement processed.

0.04seconds
```

8)

```

1 declare
2 customer customers%rowtype;
3 c_name customers.customer_name%type;
4 begin
5 select * into customer from customers where customer_id=101;
6 c_name:=customer.customer_name;
7 dbms_output.put_line('CUSTOMER NAME: '||c_name);
8 end;
9
```

```

Results Explain Describe Saved SQL History

CUSTOMER NAME: John Doe

Statement processed.

0.04seconds
```