**Database Management Systems**

**Lab Cycle-III (PL/SQL Basic Concepts)**

1. Write a PL/SQL program to input two numbers and display the total and average of these numbers.

|  |
| --- |
| SET SERVEROUTPUT ON;  DECLARE  a number;  b number;  total number;  average number;  BEGIN  a:= &a;  b:= &b;  total := a + b;  average := total/2;  dbms\_output.put\_line(' Sum of '|| a || ' and ' || b || ' is : ' || total);  dbms\_output.put\_line(' Average is : ' || average);  END; |

1. Write a PL/SQL block to accept a year and check whether it is a leap year or not?

|  |
| --- |
| SET SERVEROUTPUT ON;  DECLARE  year NUMBER := &year;  BEGIN  IF MOD(year,4)=0 AND MOD(year,100)!=0 OR MOD(year,400)=0 THEN  dbms\_output.Put\_line(year || ' is a leap year ');  ELSE  dbms\_output.Put\_line(year || ' is not a leap year.');  END IF;  END; |

1. Write a program to input the salary and working experience of employee and calculate the bonus as 10% of salary. Give $500 Extra bonus to those who’s working experience More than 10 years.

|  |
| --- |
| SET SERVEROUTPUT ON;  DECLARE  sal NUMBER := &sal;  exp number := &exp;  bonus number;    BEGIN  IF exp<=10 THEN  bonus := (sal \* 0.1) + sal;  ELSE  bonus := (sal \* 0.1) + sal + 500;  END IF;  dbms\_output.Put\_line('bonus calculated is '|| bonus);  END; |

1. Write a PL/SQL program to input the Basic Salary and calculate the HRA, DA and Net Salary as per:

BASIC HRA DA

>15000 12% 8%

>12000 10% 6%

>9000 7% 4%

OTHERS 5% $200

|  |
| --- |
| SET SERVEROUTPUT ON;  DECLARE  basic NUMBER := &basic;  hra number;  da number;  net\_sal number;  BEGIN  if basic>15000 then  hra := (basic \* .12);  da := (basic \* .08);  elsif basic>12000 then  hra := (basic \* .10);  da := (basic \* .06);  elsif basic > 9000 then  hra := (basic \* .07);  da := (basic \* .04);  else  hra := (basic \* .05);  da := 500;  end if;  net\_sal := basic + hra + da;  dbms\_output.Put\_line('Net Salary is '|| net\_sal);  END; |

1. Program to input principal amount, time. If time more than 10 years, calculate the simple interest with 8% interest otherwise with 6%.

|  |
| --- |
| SET SERVEROUTPUT ON;  DECLARE  si\_amt number := &si\_amt;  si\_time number := &si\_time;  simple\_int number;  BEGIN  if si\_time>10 then  simple\_int := (si\_amt \* si\_time \* 8) / 100;  else  simple\_int := (si\_amt \* si\_time \* 6) / 100;  end if;  dbms\_output.Put\_line('Simple Interest is '|| simple\_int);  END; |

**SELECT, INSERT AND DELETE:**

1. Write the Code to input the employee’s number and print the name and salary of that employee.

|  |
| --- |
| SET SERVEROUTPUT ON;  DECLARE  empid number:= &empid;  e\_id employee.eid%type;  ename employee.e\_name%type;  esalary employee.salary%type;    BEGIN  SELECT eid,e\_name,salary INTO e\_id,ename,esalary from employee where eid = empid;  dbms\_output.put\_line(rpad('Employee ID',15)||rpad('Name',25) || 'Salary %');  dbms\_output.Put\_line('---------------------------------------------------');  dbms\_output.put\_line(rpad(e\_id,15) ||rpad(ename,25)|| esalary);  END; |

1. Write a PL/SQL block to print the highest paid and lowest paid employee from employee table.

|  |
| --- |
| SET SERVEROUTPUT ON;  DECLARE  e\_id employee.eid%type;  ename employee.e\_name%type;  esalary employee.salary%type;  CURSOR e\_cursor1 IS  SELECT eid,ename,salary from employee order by salary desc;  CURSOR e\_cursor2 IS  SELECT eid,ename,salary from employee order by salary ;  BEGIN  OPEN e\_cursor;  dbms\_output.put\_line(rpad('Employee ID',15)||rpad('Name',25) || 'Salary %');  dbms\_output.Put\_line('---------------------------------------------------');  FETCH e\_cursor1 INTO e\_id,ename,esalary;  dbms\_output.put\_line(rpad(e\_id,15) ||rpad(ename,25)|| esalary);  FETCH e\_cursor1 INTO e\_id,ename,esalary;  dbms\_output.put\_line(rpad(e\_id,15) ||rpad(ename,25)|| esalary);  CLOSE e\_cursor;  END; |

1. Write the PL/SQL code to input the employee’s number and increase the salary by 10% if his salary < 5000 otherwise delete the record.

|  |
| --- |
| SET SERVEROUTPUT ON;  DECLARE  empid number:= &empid;  e\_id employee.eid%type;  ename employee.e\_name%type;  esalary employee.salary%type;  CURSOR e\_cursor IS  SELECT eid,e\_name,salary from employee where eid = empid;  BEGIN  OPEN e\_cursor;  FETCH e\_cursor INTO e\_id,ename,esalary;  if esalary > 60000 then  esalary := esalary + (esalary \* 0.1);  update employee set salary = esalary where eid = empid;  dbms\_output.put\_line('Salary increased by 10%');  else  delete from employee where eid = empid;  dbms\_output.put\_line('Record Deleated');  END if;  CLOSE e\_cursor;  END; |

1. Write a PL/SQL block to delete all the rows of a particular department from the table employee where deptno is accepted from the user.

|  |
| --- |
| SET SERVEROUTPUT ON;  DECLARE    depid number:= &depid;  e\_id employee.eid%type;  CURSOR e\_cursor IS  SELECT eid from employee where dept\_id = depid;  BEGIN  OPEN e\_cursor;  LOOP  FETCH e\_cursor INTO e\_id;  EXIT WHEN e\_cursor%NOTFOUND;  delete from employee where eid = e\_id;  dbms\_output.put\_line('Record Deleated');  END LOOP;  CLOSE e\_cursor;  END; |

1. Write PL/SQL code to insert the record in department table.

|  |
| --- |
| SET SERVEROUTPUT ON;  DECLARE  e\_id employee.eid%type;  ename employee.e\_name%type ;  sal employee.salary%type ;  des employee.designation%type;  dep employee.dept\_id%type;  BEGIN  insert INTO employee (eid,e\_name,salary,designation,dept\_id)  values (&e\_id,'&ename',&sal,'&des',&dep);  commit;  dbms\_output.put\_line('Record Inserted');  END; |

**EXCEPTIONS:**

1. Write PL/SQL script to input salary amount and display the Employee Name earning same salary amount. Use NO\_DATA\_FOUND and TOO\_MANY ROWS Exception.

|  |
| --- |
| SET SERVEROUTPUT ON;  DECLARE  sal employee.salary%type := &sal;  e\_id employee.eid%type;  ename employee.e\_name%type;  BEGIN  SELECT eid,e\_name INTO e\_id,ename FROM employee where salary = sal;  DBMS\_OUTPUT.PUT\_LINE ('Employee ID: '|| e\_id);  DBMS\_OUTPUT.PUT\_LINE ('Employee Name ' || ename);  EXCEPTION  WHEN no\_data\_found THEN  dbms\_output.put\_line('No Such Employee Found!');  WHEN others THEN  dbms\_output.put\_line('Too Many Lines Exception!');  END; |

1. Write PL/SQL script that traps ZERO\_DIVIDE exception when a number is divided by other number. And also raised user define exception if number2 greater than number1.

|  |
| --- |
| SET SERVEROUTPUT ON;  DECLARE  num1 number := &num1;  num2 number := &num2;  res number ;  my\_exception EXCEPTION;  BEGIN  if num2 > num1 then  RAISE my\_exception;  else  res := num1 / num2;  end if;  dbms\_output.put\_line('Division of the number is : '|| res);  EXCEPTION  WHEN my\_exception THEN  dbms\_output.put\_line('Dividion not possible Number 2 is greater than Number 1');  WHEN ZERO\_DIVIDE THEN  dbms\_output.put\_line('Division by Zero not possible!');  END; |