8. PL/SQL PROGRAM TO PREPARE PAYROLL USING FUNCTIONS

AIM

To write a DBMS program to prepare reports for a payroll application using functions.

ALGORITHM

Step – I : In the notepad window, create a PL/SQL function and save it in a file (say E:\SKS\MC9217\Program_7\Pay_Calc.txt).

Step – II : The function calculates the gross pay by summing up basic pay, dearness allowance (29% of basic pay), house rent allowance, city commercial allowance (Rs. 300), special pay, and arrear. It also calculates deductions by summing up loss of pay, salary advance, provident fund, group insurance, staff welfare, and income tax. The function then returns the net pay by subtracting deductions from gross pay.

Step – III : Execute the function as shown below:

```
SQL>  @ E:\SKS\MC9217\Program_7\Pay_Calc.txt

Function created.
```

Step – IV : Type ed in the SQL Editor. Key in the PL/SQL procedure and save it in a file (say E:\SKS\MC9217\Program_7\PLSQL_Pay.txt).

Step – V : The PL / SQL procedure accepts employee name, basic pay, house rent allowance, special pay, arrear, loss of pay, salary advance, provident fund, group insurance, staff welfare, and income tax as input. It then invokes the function Pay_Calc using these values.

Step – VI : The procedure prepares a payroll report by using the input values and the calculated gross pay and net pay.

Step – VII : Execute the PL / SQL procedure as shown below:

```
SQL>  @ E:\SKS\MC9217\Program_7\PLSQL_Pay.txt
```

Step – VIII: Execute the following command for the dbms_output.put_line to take effect.

```
SQL>  set serveroutput on
```

PL / SQL PROCEDURE

Pay_Calc.txt

```
create or replace function Pay_Calc (bp number, hra number, sp
  number, arr number, lop number, sa number, pf number, gi number,
  sw number, it number) return number is
 gp number;
 ded number;
begin
 dbms_output.put_line('Dearness Allowance            : Rs. ' ||
  to_char(round(bp*29/100,0)));
 dbms_output.put_line('City Commercial Allowance   : Rs. 300');
 gp := bp + round(bp * 29/100,0) + hra + 300 + sp + arr;
 dbms_output.put_line('-----------------------------');
 dbms_output.put_line('Gross Pay                     : Rs. ' ||
  to_char(gp));
 dbms_output.put_line('-----------------------------');
 ded := lop + sa + pf + gi + sw + it;
 return (gp - ded);
end;
/
```

PLSQL_Pay.txt

```
declare
 name varchar2(30);
 bp number;
 hra number;
 sp number;
 arr number;
 lop number;
 sa number;
 pf number;
 gi number;
 sw number;
 it number;
 np number;
begin
 name := '&Employee_Name';
 bp := &Basic_Pay;
 hra := &House_Rent_Allowance;
 sp := &Special_Pay;
 arr := &Arrear;
 lop := &Loss_of_Pay;
 sa := &Salary_Advance;
 pf := &Provident_Fund;
 gi := &Group_Insurance;
 sw := &Staff_Welfare;
 it := &Income_tax;
 dbms_output.put_line('Employee Name               : ' || name);
 dbms_output.put_line('-----------------------------');
 dbms_output.put_line('Earnings');
 dbms_output.put_line('-----------------------------');
 dbms_output.put_line('Basic Pay           : Rs. ' || to_char(bp));
 dbms_output.put_line('House Rent Allowance     : Rs. ' || to_char(hra));
 dbms_output.put_line('Special Pay         : Rs. ' || to_char(sp));
 dbms_output.put_line('Arrear            : Rs. ' || to_char(arr));
 np := Pay_Calc (bp, hra, sp, arr, lop, sa, pf, gi, sw, it);
 dbms_output.put_line('Deductions');
 dbms_output.put_line('-----------------------------');
 dbms_output.put_line('Loss of Pay       : Rs. ' || to_char(lop));
 dbms_output.put_line('Salary Advance      : Rs. ' || to_char(sa));
 dbms_output.put_line('Provident Fund      : Rs. ' || to_char(pf));
 dbms_output.put_line('Group Insurance     : Rs. ' || to_char(gi));
 dbms_output.put_line('Staff Welfare       : Rs. ' || to_char(sw));
 dbms_output.put_line('Income Tax          : Rs. ' || to_char(it));
 dbms_output.put_line('-----------------------------');
 dbms_output.put_line('Net Pay             : Rs. ' || to_char(np));
 dbms_output.put_line('-----------------------------');
end;
/
```

OUTPUT

```
SQL> @ E:\SKS\MC9217\Program_7\PLSQL_Pay.txt
     Enter value for employee_name: S.K. Saravanan
     old  15:  name := '&Employee_Name';
     new  15:  name := ‘S.K. Saravanan’;
     Enter value for basic_pay: 12413
```

```
old  16:  bp := &Basic_Pay;
new  16:  bp := 12413;
Enter value for house_rent_allowance: 1500
old  17:  hra := &House_Rent_Allowance;
new  17:  hra := 1500;
Enter value for special_pay: 0
old  18:  sp := &Special_Pay;
new  18:  sp := 0;
Enter value for arrear: 0
old  19:  arr := &Arrear;
new  19:  arr := 0;
Enter value for loss_of_pay: 0
old  20:  lop := &Loss_of_Pay;
new  20:  lop := 0;
Enter value for salary_advance: 0
old  21:  sa := &Salary_Advance;
new  21:  sa := 0;
Enter value for provident_fund: 780
old  22:  pf := &Provident_Fund;
new  22:  pf := 780;
Enter value for group_insurance: 150
old  23:  gi := &Group_Insurance;
new  23:  gi := 150;
Enter value for staff_welfare: 20
old  24:  sw := &Staff_Welfare;
new  24:  sw := 20;
Enter value for income_tax: 0
old  25:  it := &Income_tax;
new  25:  it := 0;
Employee Name               : S.K. Saravanan
-----------------------------
Earnings
-----------------------------
Basic Pay                   : Rs. 12413
House Rent Allowance        : Rs. 1500
Special Pay                 : Rs. 0
Arrear                      : Rs. 0
Dearness Allowance          : Rs. 3600
City Commercial Allowance   : Rs. 300
-----------------------------
Gross Pay                   : Rs. 17813
-----------------------------
Deductions
-----------------------------
Loss of Pay                 : Rs. 0
Salary Advance              : Rs. 0
Provident Fund              : Rs. 780
Group Insurance             : Rs. 150
Staff Welfare               : Rs. 20
Income Tax                  : Rs. 0
-----------------------------
Net Pay                     : Rs. 16863
-----------------------------

PL/SQL procedure successfully completed.
```

RESULT:

Thus, the above program was executed successfully.