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

num\_small NUMBER := 8;

num\_large NUMBER := 5;

num\_temp NUMBER;

BEGIN

IF num\_small > num\_large THEN

num\_temp := num\_small;

num\_small := num\_large;

num\_large := num\_temp;

END IF;

DBMS\_OUTPUT.PUT\_LINE ('num\_small = '||num\_small);

DBMS\_OUTPUT.PUT\_LINE ('num\_large = '||num\_large);

END;

/

Copy

Sample Output:

num\_small = 5

num\_large = 8

Write a PL/SQL program to count the number of employees in department 50 and check whether this department have any vacancies or not. There are 45 vacancies in this department.

SET SERVEROUTPUT ON

DECLARE

tot\_emp NUMBER;

BEGIN

SELECT Count(\*)

INTO tot\_emp

FROM employees e

join departments d

ON e.department\_id = d.department\_id

WHERE e.department\_id = 50;

dbms\_output.Put\_line ('The employees are in the department 50: '

||To\_char(tot\_emp));

IF tot\_emp >= 45 THEN

dbms\_output.Put\_line ('There are no vacancies in the department 50.');

ELSE

dbms\_output.Put\_line ('There are some vacancies in department 50.');

END IF;

END;

/

Copy

Sample Output:

The employees are in the department 50: 45

There are no vacancies in the department 50.

PL/SQL procedure successfully completed.

Write a PL/SQL program to check whether a date falls on weekend i.e. SATURDAY or SUNDAY.

**Sample Solution:**

**PL/SQL Code:**

DECLARE

dt1 DATE := TO\_DATE('&new\_dt', 'DD-MON-YYYY');

get\_day VARCHAR2(15);

BEGIN

get\_day := RTRIM(TO\_CHAR(dt1, 'DAY'));

IF get\_day IN ('SATURDAY', 'SUNDAY') THEN

dbms\_output.new\_line;

DBMS\_OUTPUT.PUT\_LINE

('The day of the given date is '||get\_day||' and it falls on weekend');

ELSE

dbms\_output.new\_line;

DBMS\_OUTPUT.PUT\_LINE ('The day of the given date is '||get\_day||' and it does not fall on the weekend');

END IF;

DBMS\_OUTPUT.PUT\_LINE ('Execution done successfully.');

END;

Write a PL/SQL program to display the description against a grade.

DECLARE

grd CHAR(1);

BEGIN

-- Accept value for grade

grd := '&new\_grd';

IF grd = 'A' THEN

dbms\_output.Put\_line('Your Grade is: Outstanding');

ELSIF grd = 'B' THEN

dbms\_output.Put\_line('Your Grade is: Excellent');

ELSIF grd = 'C' THEN

dbms\_output.Put\_line('Your Grade is: Very Good');

ELSIF grd = 'D' THEN

dbms\_output. Put\_line('Your Grade is: Average');

ELSIF grd = 'F' THEN

dbms\_output.Put\_line('Your Grade is: Poor');

ELSE

dbms\_output.Put\_line('No such grade in the list.');

END IF;

END;

Write a program in PL/SQL to insert records from one table to another.

DROP TABLE emp\_temp;

CREATE TABLE emp\_temp (

emp\_id NUMBER,

emp\_email VARCHAR2(40)

);

DECLARE

number\_of\_emp NUMBER;

BEGIN

SELECT COUNT(employee\_id) INTO number\_of\_emp

FROM employees;

FOR i IN 1..number\_of\_emp LOOP

INSERT INTO emp\_temp (emp\_id, emp\_email)

VALUES(i, 'not available now');

END LOOP;

END;

/

**Date Functions** :-

**1) SYSDATE** :- Displays the system date for a system

SQL> SELECT SYSDATE FROM DUAL;

SQL> SELECT TO\_CHAR (SYSDATE, 'DD-MON-YYYY HH:MI:SS') FROM dual;

**2) NEXT\_DAY (D, DAY)** :- Displays next date on DAY after date D.

Ex: Display date on Thu after 20th Feb, 2010.

SQL> SELECT NEXT\_DAY (’20-FEB-2010’, ‘THU’) FROM DUAL;

**3) ADD\_MONTHS (D, N)** :- Returns a date after adding a specified day D with specified

number of months N.

Ex: Display SID, Day of Reservation by adding 20 months to given day.

SQL> SELECT SID, DAY, ADD\_MONTHS (DAY, 20) FROM RESERVES

**4) LAST\_DAY(D)** :- Returns the date corresponding to last day of the month.

Ex: Display Sname, Day of Reservation and date corresponding to last date of the month.

SQL> SELECT S.SNAME, DAY, LAST\_DAY (DAY) FROM SAILORS S, RESERVES R

WHERE S.SID = R.SID;

**5) MONTHS\_BETWEEN (D1, D2)** :- Returns number of months between given two

dates D1 & D2.

Ex: Display SID, Day of Reservation and months between System Date & day of reservation.

SQL> SELECT SID, DAY, MONTHS\_BETWEEN (SYSDATE, DAY) FROM RESERVES;