|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing drawing, stop, room  Description automatically generated | Database Management Systems  Practical #10 | | |
|  |  |  |  |
| **Name** | Romulus Darwin | **Roll Number** | 21302A0008 |
| **Subject/Course:** | Database Management Systems / BSc IT | | |
| **Practical 9** | PLSQL Questions | | |
|  | | | |
| **Write PL/SQL program with Output as Screen shots for the following Questions.**   1. Write a PL/SQL block to print the sum of individual digits of a given number and check whether the given number is Armstrong or not. (Use simple/basic loop) 2. Write a PL/SQL block to check whether the given number is positive or negative or neutral. Also find factorial value of the entered number. 3. Write a PL/SQL block that creates procedure to perform Volume of Cylinder by accepting radius and height as input parameters and store the result in an output parameter. The output parameter is to be printed at the time of execution. 4. Write two procedures area\_rect () and perimeter-rect() that accept length and breadth of rectangle as parameters and calculate area and perimeter of the rectangle respectively. 5. Write a PL/SQLblock to find the sum of square of n numbers and the sum of the individual digits of a given number. (Use while loop) 6. Write a PL/SQLblock to find the sum of cube of n numbers and the reverse of a given number. (Use basic/simple loop). | | | |
| 1. **Program:**   declare  n number:=407;  s number:=0;  r number;  len number;  m number;  begin  m:=n;  len:=length(to\_char(n));  while n>0  loop  r:=mod(n,10);  s:=s+power(r,len);  n:=trunc(n/10);  end loop;  if m=s  then  dbms\_output.put\_line('armstrong number');  else  dbms\_output.put\_line('not armstrong number');  end if;  end;  /     1. **Program**   DECLARE  NO NUMBER;  BEGIN NO:=&NO;  IF NO < 0 THEN  DBMS\_OUTPUT.PUT\_LINE('NEGATIVE NUMBER');  ELSIF NO > 0 THEN  DBMS\_OUTPUT.PUT\_LINE('POSITIVE NUMBER');  ELSE  DBMS\_OUTPUT.PUT\_LINE('EQUAL TO ZERO');  END IF;  END;  /    **3 program**  **CREATE OR REPLACE PROCEDURE VOL(R IN NUMBER,H IN NUMBER)**  **IS**  **V NUMBER;**  **BEGIN**  **V:=3.14\*R\*R\*H;**  **DBMS\_OUTPUT.PUT\_LINE('VOL OF CYLINDER IS ' || V);**  **END;**  **/**  **DECLARE**  **RADIUS NUMBER; HEIGHT NUMBER;**  **BEGIN**  **RADIUS:=&RADIUS; HEIGHT:=&HEIGHT;**  **VOL(RADIUS,HEIGHT);**  **END;**  **/**    **4 Program**  CREATE OR REPLACE PROCEDURE AR(L IN NUMBER,B IN NUMBER)  IS  A NUMBER;  BEGIN  A:=L\*B;  DBMS\_OUTPUT.PUT\_LINE('AREA OF RECTANGLE IS ' || A);  END;  /  CREATE OR REPLACE PROCEDURE PER(LE IN NUMBER,BR IN NUMBER)  IS  P NUMBER;  BEGIN  P:=2\*LE+2\*BR;  DBMS\_OUTPUT.PUT\_LINE('PERIMETER OF RECTANGLE IS ' || P);  END;  /  DECLARE  LENGTH NUMBER; BREADTH NUMBER;  BEGIN  LENGTH:=&LENGTH; BREADTH:=&BREADTH;  AR(LENGTH,BREADTH);  PER(LENGTH,BREADTH);  END;  /    **5 program**  declare     n number;      s number:=0;      r number:=0;      m number;      len number;     c number:=1; begin     n:=&n;     m:=n;      len:=length(to\_char(n));      while n>0      loop          r:=mod(n,10);          s:=s+power(r,2);          n:=trunc(n/10);      end loop;            dbms\_output.put\_line('sum of ' || m || ' = ' || s);     while(c <= m)     loop     r:=r+power(c,2);     c:= c+1;     end loop;     dbms\_output.put\_line('sum of square of numbers upto ' || m || ' = ' || r);       end; / | | | |
| **6 program**  declare     n number;      s number:=0;      r number:=0;      m number;      len number;     c number:=1;     sum\_of\_cube number := 0; begin     n:=&n;     m:=n;      len:=length(to\_char(n));           loop          r:=mod(n,10);          s:=(s \* 10) + r;          n:=trunc(n/10);      exit when n <= 0;     end loop;            dbms\_output.put\_line('Reverse of ' || m || ' = ' || s);     loop     sum\_of\_cube:=sum\_of\_cube+power(c,3);     c:= c+1;     exit when c > m;     end loop;     dbms\_output.put\_line('sum of cube of numbers upto ' || m || ' = ' || sum\_of\_cube);       end; / | | | |
|  | | | |