

DBMS Assignment 4

Swapnadeep Mishra

IT UG2

SEC-A3

ROLL:002211001115

1) Write a PL/SQL code to print Today is fall on weekend or weekdays using if else statement.

```
begin
if to_char(sysdate,'dy') in ('SAT','SUN') then
    dbms_output.put_line('Weekend');
else
    dbms_output.put_line(to_char(sysdate,'dy'));
end if;
end;
```

Schema

Quick SQL

My Scripts

My Tutorials

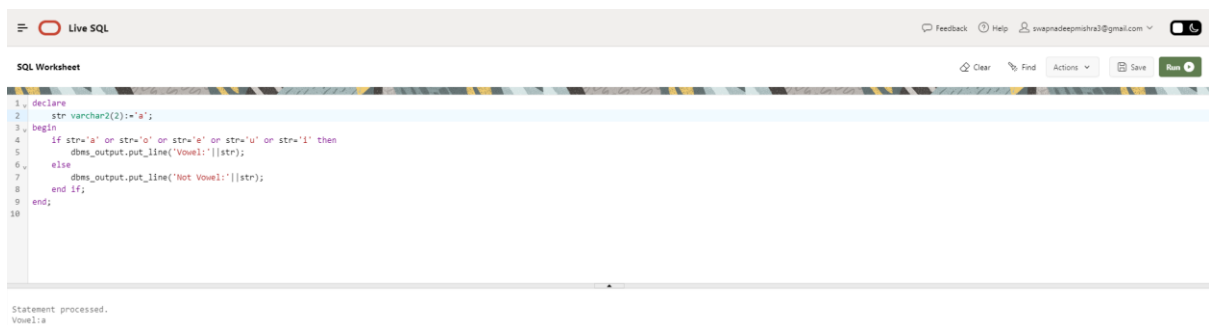
Code Library

```
44 begin
45 if to_char(sysdate,'dy') in ('SAT','SUN') then
46     dbms_output.put_line('Weekend');
47 else
48     dbms_output.put_line('Weekday');
49 end if;
50 end;
51
52
53
```

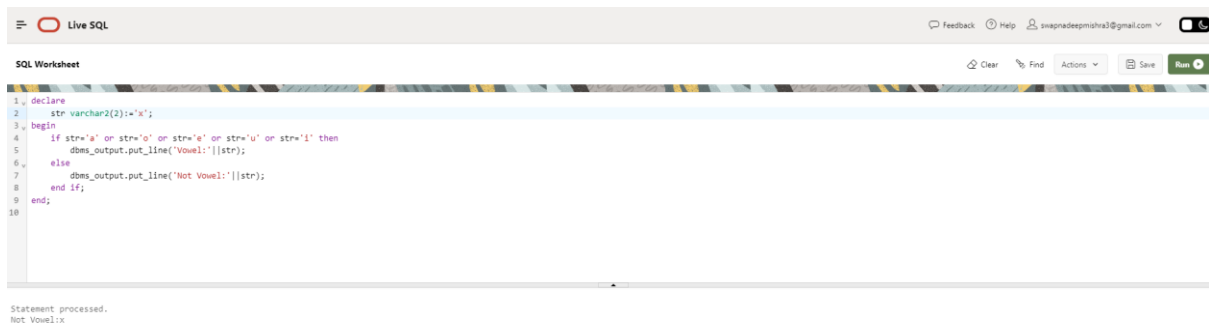
Statement processed.
Weekday

2) Write a PL/SQL code to check that an inputted a single character is vowel or not .If vowel then display which vowel it is.

```
declare
    str varchar2(2):='x';
begin
    if str='a' or str='o' or str='e' or str='u' or str='i' then
        dbms_output.put_line('Vowel:'||str);
    else
        dbms_output.put_line('Not Vowel:'||str);
    end if;
end;
```



The screenshot shows a web-based SQL editor titled "Live SQL". The code area contains the PL/SQL block for checking vowels. The output area at the bottom displays "Statement processed." followed by "Vowel:a".



The screenshot shows the same "Live SQL" interface. The code area is identical. The output area at the bottom displays "Statement processed." followed by "Not Vowel:x".

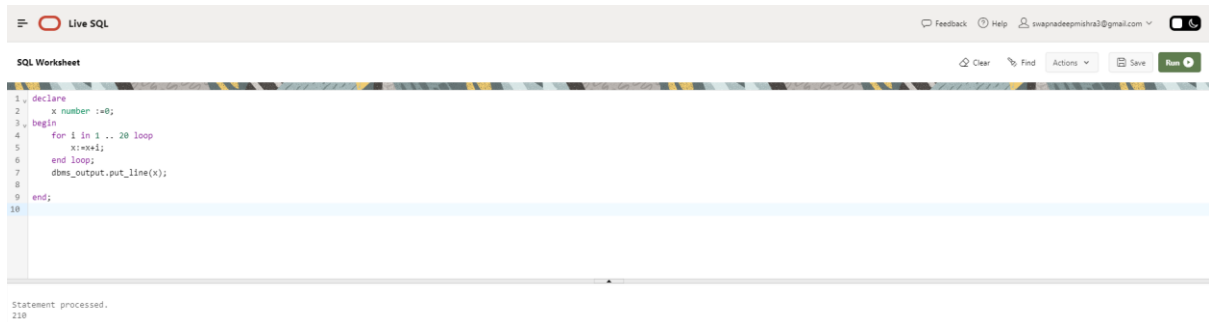
3) Write a PL/SQL code block to find out the sum of first twenty natural numbers (1+2+3+4+5+6+7+8+9+10+-----+20 this series).

```
declare
```

```

        x number :=0;
begin
    for i in 1 .. 20 loop
        x:=x+i;
    end loop;
    dbms_output.put_line(x);
end;

```



4) Write a PL/SQL block that will ask for two numbers and one operand (+, -, *, /). Then it will calculate and display the result.

```

declare
    num1 number :=100;
    num2 number :=200;
    op varchar2(2) :='*';

begin
    if op='+' then
        dbms_output.put_line(num1+num2);
    elsif op='-' then
        dbms_output.put_line(num1-num2);
    elsif op='*' then
        dbms_output.put_line(num1*num2);
    elsif op='/' then
        dbms_output.put_line(num1/num2);
    end if;
end;

```

Live SQL

FeedbackHelpswapnadeepmishra3@gmail.com

SQL Worksheet

ClearFindActionsSaveRun

```
1 declare
2   num1 number :=100;
3   num2 number :=200;
4   op varchar2(2) := '+';
5
6 begin
7   if op='+' then
8     dbms_output.put_line(num1+num2);
9   elsif op='-' then
10    dbms_output.put_line(num1-num2);
11  elsif op='*' then
12    dbms_output.put_line(num1*num2);
13  elsif op='/' then
14    dbms_output.put_line(num1/num2);
15  end if;
16 end;
```

Statement processed.
20000

Live SQL

FeedbackHelpswapnadeepmishra3@gmail.com

SQL Worksheet

ClearFindActionsSaveRun

```
1 declare
2   num1 number :=100;
3   num2 number :=200;
4   op varchar2(2) := '/';
5
6 begin
7   if op='+' then
8     dbms_output.put_line(num1+num2);
9   elsif op='-' then
10    dbms_output.put_line(num1-num2);
11  elsif op='*' then
12    dbms_output.put_line(num1*num2);
13  elsif op='/' then
14    dbms_output.put_line(num1/num2);
15  end if;
16 end;
```

Statement processed.
.5

Live SQL

FeedbackHelpswapnadeepmishra3@gmail.com

SQL Worksheet

ClearFindActionsSaveRun

```
1 declare
2   num1 number :=100;
3   num2 number :=200;
4   op varchar2(2) := '-';
5
6 begin
7   if op='+' then
8     dbms_output.put_line(num1+num2);
9   elsif op='-' then
10    dbms_output.put_line(num1-num2);
11  elsif op='*' then
12    dbms_output.put_line(num1*num2);
13  elsif op='/' then
14    dbms_output.put_line(num1/num2);
15  end if;
16 end;
```

Statement processed.
-100

Live SQL

FeedbackHelpswapnadeepmishra3@gmail.com

SQL Worksheet

ClearFindActionsSaveRun

```
1 declare
2   num1 number :=100;
3   num2 number :=200;
4   op varchar2(2) := '*';
5
6 begin
7   if op='+' then
8     dbms_output.put_line(num1+num2);
9   elsif op='-' then
10    dbms_output.put_line(num1-num2);
11  elsif op='*' then
12    dbms_output.put_line(num1*num2);
13  elsif op='/' then
14    dbms_output.put_line(num1/num2);
15  end if;
16 end;
```

Statement processed.
300

5) Write a PL/SQL code block to display a number in a reverse way.

```

declare
    num integer :=123;
    rev integer :=0;
    rem integer;
begin
    while num>0 loop
        rem:=mod(num,10);
        rev:= 10*rev+rem;
        num:= num/10;
    end loop;
    dbms_output.put_line(rev);
end;

```

The screenshot shows a web browser window with a tab titled "Live SQL". The address bar shows a URL with a user profile icon and the email "svapnadeepmishra@gmail.com". The page has a header with "Feedback", "Help", and a user profile dropdown. Below the header is a section titled "SQL Worksheet" with a toolbar containing "Clear", "Find", "Actions", "Save", and "Run" buttons. The main area contains a PL/SQL block, which is identical to the one in the first block. Line numbers 1 through 13 are visible on the left. At the bottom, a status bar indicates "Statement processed." and "321".

```

1 declare
2   num integer :=123;
3   rev integer :=0;
4   rem integer;
5 begin
6   while num>0 loop
7     rem:=mod(num,10);
8     rev:= 10*rev+rem;
9     num:= num/10;
10  end loop;
11  dbms_output.put_line(rev);
12 end;
13

```

Statement processed.
321

6) Write a PL/SQL block to display the dates of this month which are Tuesday.

```

declare
    curr date:=trunc(sysdate,'mm');
    last date:=last_day(sysdate);
begin
    while curr<=last loop
        if to_char(curr,'d')=3 then
            dbms_output.put_line(curr);
        end if;
        curr:=curr+1;
    end loop;
end;

```

```

        end if;

    curr:=curr+1;
end loop;

end;

```

The screenshot shows a web-based SQL editor titled "Live SQL". The code area contains a PL/SQL program that declares a variable 'curr' and a loop that prints the day of the week for dates from the current date to the last day of the month. The execution results show the output of the program, which is a list of dates from 03-OCT-23 to 31-OCT-23.

```

1 declare
2     curr date:=trunc(sysdate,'mm');
3     last date:=last_day(sysdate);
4 begin
5     while curr<=last loop
6         if to_char(curr,'d')=3 then
7             dbms_output.put_line(curr);
8         end if;
9         curr:=curr+1;
10    end loop;
11 end;
12
13

```

Statement processed.
03-OCT-23
10-OCT-23
17-OCT-23
24-OCT-23
31-OCT-23

7) Write a program in PL/SQL to print the prime numbers between 1 to 50.

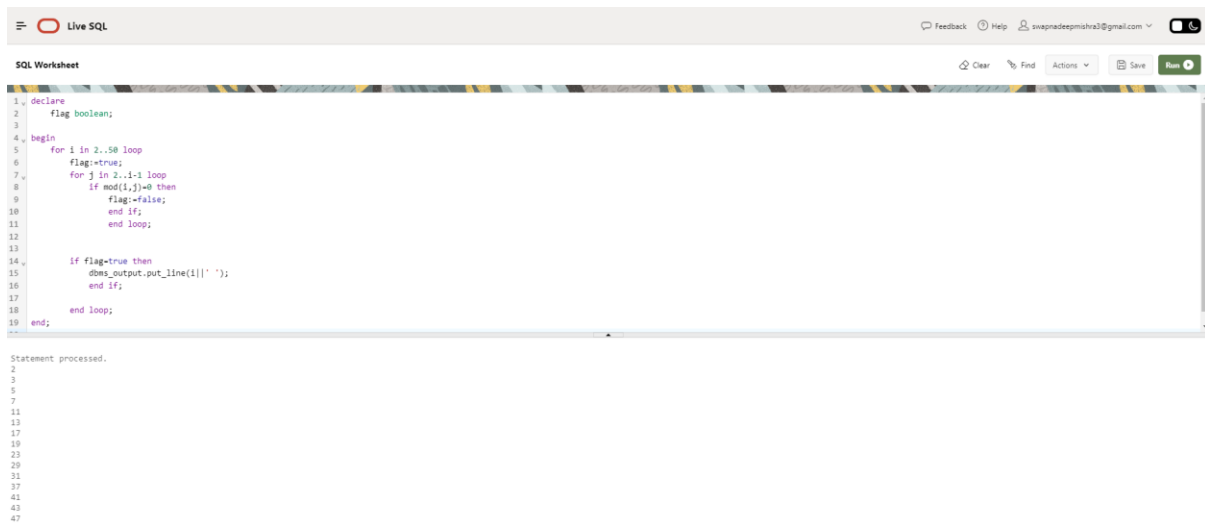
```

declare
    flag boolean;
begin
    for i in 2..50 loop
        flag:=true;
        for j in 2..i-1 loop
            if mod(i,j)=0 then
                flag:=false;
            end if;
        end loop;

        if flag=true then
            dbms_output.put_line(i||' ');
        end if;
    end loop;

```

end;



The screenshot shows a web-based SQL editor interface. At the top, there's a header with the 'Live SQL' logo and user information. Below the header, the editor area contains a PL/SQL program. The program declares a boolean flag, enters a loop for i from 2 to 50, and inside that, a loop for j from 2 to i-1. It checks if i is divisible by j, toggles the flag, and prints the flag's value when i is prime. The output pane below shows the execution results, with the flag value 'true' printed for prime numbers 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, and 47.

```
1 declare
2   flag boolean;
3
4 begin
5   for i in 2..50 loop
6     flag:=true;
7     for j in 2..i-1 loop
8       if mod(i,j)=0 then
9         flag:=false;
10        end if;
11      end loop;
12
13      if flag=true then
14        dbms_output.put_line(i||' ');
15      end if;
16    end loop;
17  end;
18 end;
```

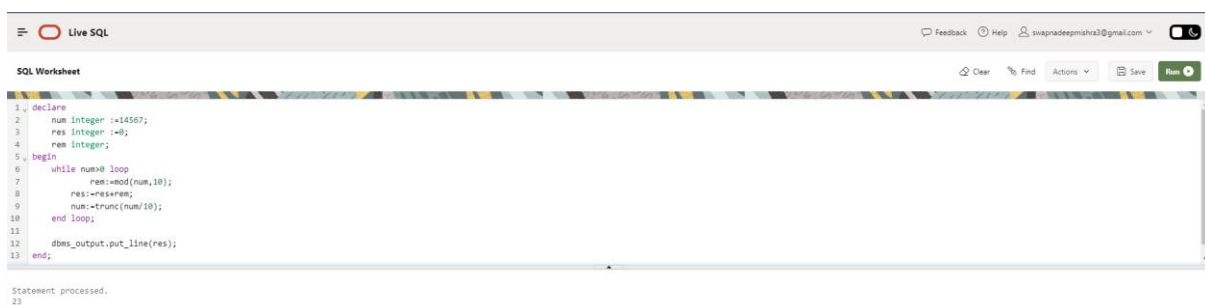
Statement processed.

2
3
5
7
11
13
17
19
23
29
31
37
41
43
47

8) Write a program in PL/SQL to print the sum of digits of a number [eg: 635=14].

```
declare
  num integer :=14567;
  res integer :=0;
  rem integer;
begin
  while num>0 loop
    rem:=mod(num,10);
    res:=res+rem;
    num:=trunc(num/10);
  end loop;

  dbms_output.put_line(res);
end;
```



The screenshot shows the same Live SQL interface. The editor contains a PL/SQL program that calculates the sum of digits of the number 14567. It uses a while loop to repeatedly extract the last digit (using mod) and add it to a running total (res), then removes the last digit (using trunc). The output pane shows the final result of 23.

```
1 declare
2   num integer :=14567;
3   res integer :=0;
4   rem integer;
5 begin
6   while num>0 loop
7     rem:=mod(num,10);
8     res:=res+rem;
9     num:=trunc(num/10);
10  end loop;
11
12  dbms_output.put_line(res);
13 end;
```

Statement processed.

23