PL/SQL Lab Assignments

Assignment -1

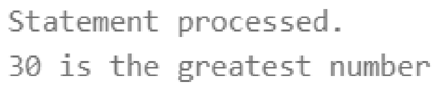
1) WAP to find the greatest of three numbers.

STATEMENT:

```
declare
a int := 30;
b int := 10;
c int := 20;
begin
if a>b then
if a>c then
dbms_output.put_line(a || ' is the greatest number');
else
dbms_output.put_line(b || ' is the greatest number');
end if;
else if b>c then
dbms_output.put_line(b || ' is the greatest number');
else
dbms_output.put_line(c || ' is the greatest number');
end if;
end if;
```

```
end;
```

OUTPUT:



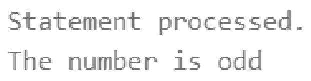
```
Statement processed.
30 is the greatest number
```

2) WAP to check whether number is odd or even.

STATEMENT:

```
declare

num int := 21;

begin

if mod(num,2)=0 then

dbms_output.put_line('The number is even');

else

dbms_output.put_line('The number is odd');

end if;

end;
```

OUTPUT:



```
Statement processed.
The number is odd
```

3) WAP to find the grade. Consider the following:

Marks > 80 A grade

Marks >70 B grade

Marks >50 C grade

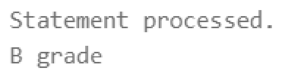
Marks > 40 D grade

Marks < 40 E grade

STATEMENT:

```
declare
marks int := 75;
begin
if marks>80 then
dbms_output.put_line('A grade');
else if marks>70 then
dbms_output.put_line('B grade');
else if marks>50 then
dbms_output.put_line('C grade');
else if marks>40 then
dbms_output.put_line('D grade');
else
dbms_output.put_line('E grade');
end if;
end if;
end if;
end if;
end;
```

OUTPUT:



```
Statement processed.
B grade
```

4) WAP to print the table of a given number. (use for loop)

STATEMENT:

declare

num int := 7;

begin

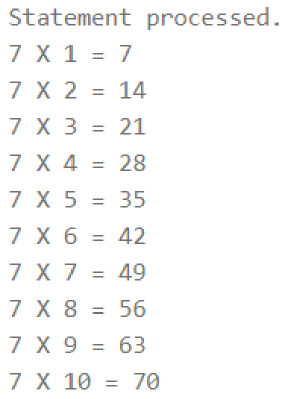
for i in 1..10 loop

dbms_output.put_line(num || ' X ' || i || ' = ' || (num*i));

end loop;

end;

OUTPUT:



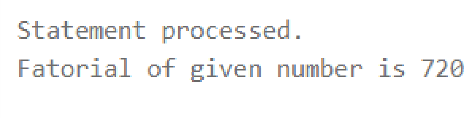
```
Statement processed.
7 X 1 = 7
7 X 2 = 14
7 X 3 = 21
7 X 4 = 28
7 X 5 = 35
7 X 6 = 42
7 X 7 = 49
7 X 8 = 56
7 X 9 = 63
7 X 10 = 70
```

5) WAP to find out the factorial of a given number. (use while loop)

STATEMENT:

```
declare
num int := 6;
ans int := 1;
begin
while num <> 0 loop
ans:=ans* num;
num:=num-1;
end loop;
dbms_output.put_line('Fatorial of given number is ' || ans);
end;
```

OUTPUT:



```
Statement processed.
Fatorial of given number is 720
```

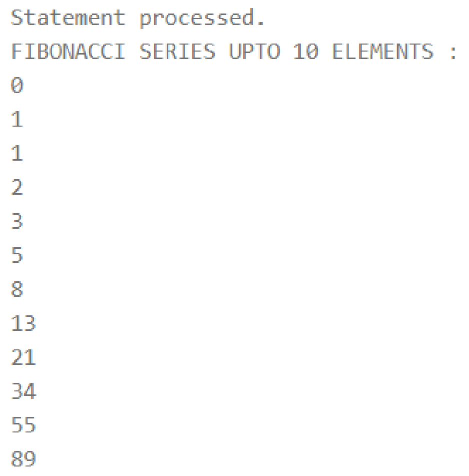
6) WAP to find out the Fibonacci series.

STATEMENT:

```
declare
e int := 10;
s1 int := 0;
s2 int := 1;
temp int := 0;
```

```
i int := 0;
begin
dbms_output.put_line('FIBONACCI SERIES UPTO 10 ELEMENTS : ');
dbms_output.put_line(s1);
dbms_output.put_line(s2);
while i<e loop
dbms_output.put_line(s1+s2);
temp:=s2;
s2:=s1+s2;
s1:=temp;
i:=i+1;
end loop;
end;
```

<u>OUTPUT:</u>



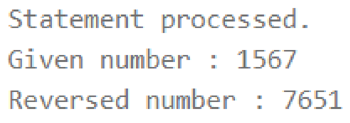
```
Statement processed.
FIBONACCI SERIES UPTO 10 ELEMENTS :
0
1
1
2
3
5
8
13
21
34
55
89
```

7) WAP to find the reverse of a number(use exit when statement)

STATEMENT:

```
declare
num number := 1567;
revnum number := 0;
begin
dbms_output.put_line('Given number : ' || num);
while num <> 0 loop
revnum := (revnum * 10) + (mod(num,10));
num := floor(num/10);
end loop;
dbms_output.put_line('Reversed number : ' || revnum);
end;
```

OUTPUT:



```
Statement processed.
Given number : 1567
Reversed number : 7651
```

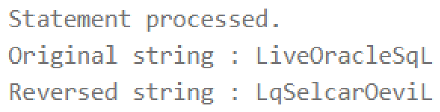
8) WAP to reverse a string.

STATEMENT:

```
declare
str1 varchar(20) := 'LiveOracleSqL';
len number;
```

```
str2 varchar(20);

begin

dbms_output.put_line('Original string : ' || str1);

len := length(str1);

while len>0 loop

str2 := str2 || substr(str1,len,1);

len:=len-1;

end loop;

dbms_output.put_line('Reversed string : ' || str2);

end;
```

<u>OUTPUT:</u>



```
Statement processed.
Original string : LiveOracleSqL
Reversed string : LqSelcarOeviL
```

PL/SQL Lab Assignment

Lab Assignment–2

Syntax:

Declare
Begin
End;
	If COND then
	Else
	End if;
		If COND then
		Elsif
		Elsif
		End if;
			Loop
			If COND then
			Exit;
			End if;
			End loop;
				Loop
				Exit when COND;
				End loop;

While COND Loop End loop;	For I in 1..10 Loop End loop;			

<u>Problems:</u>

1. PL/SQL block to update total sal for empno 100.

Eno,ename, bp,da,hra,total.

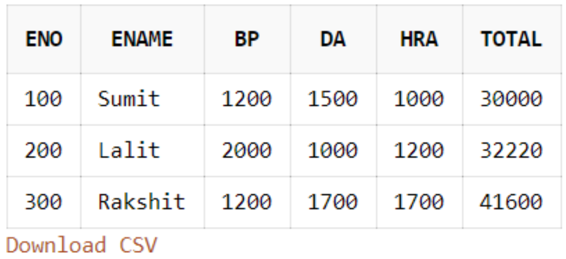
<u>STATEMENT:</u>

```
create table emp(eno number(10),ename varchar(10),bp number(10),da number(10),hra number(10),total number(10));

insert into emp values(100,'Sumit',1200,1500,1000,30000);

insert into emp values(200,'Lalit',2000,1000,1200,32220);

insert into emp values(300,'Rakshit',1200,1700,1700,41600);

select * from emp;
```

```
declare
b emp.bp%TYPE;
d emp.da%TYPE;
h emp.hra%TYPE;
t emp.total%TYPE;
begin
select bp,da,hra into b,d,h from emp where eno=100;
t := b+d+h;
update emp set total=t where eno=100;
end;
select * from emp;
```

OUTPUT:

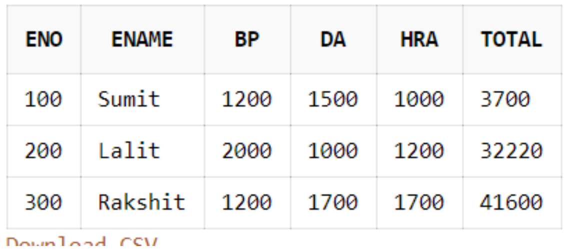
Before:



ENO	ENAME	BP	DA	HRA	TOTAL
100	Sumit	1200	1500	1000	30000
200	Lalit	2000	1000	1200	32220
300	Rakshit	1200	1700	1700	41600

Download CSV

After:



ENO	ENAME	BP	DA	HRA	TOTAL
100	Sumit	1200	1500	1000	3700
200	Lalit	2000	1000	1200	32220
300	Rakshit	1200	1700	1700	41600

Download CSV

2. PL/SQL block to calculate fine for rno 100

Rno, bookno, doi, dor, fine

Fine is rs 1 if days<7

Fine is rs 2 if days<14 and >7

Fine is rs 3 if days>14

Amount mentioned is for each day.

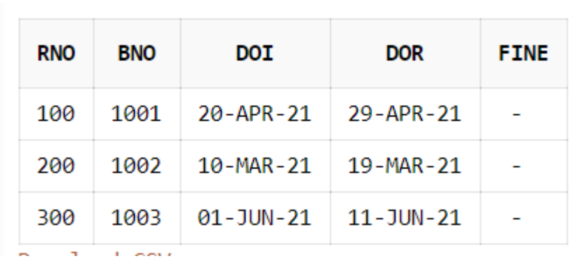
STATEMENT:

```
create table lib(rno number(10),bno number(10),doi date,dor date,fine number(10));

insert into lib values(100,1001,'20-Apr-2021','29-Apr-2021',null);

insert into lib values(200,1002,'10-Mar-2021','19-Mar-2021',null);

insert into lib values(300,1003,'01-Jun-2021','11-Jun-2021',null);

Select * from lib;


declare

days number;

f lib.fine%TYPE;

begin

select to_char(dor-doi) into days from lib where rno = 100;

if days<7 then

f:=1;

elsif days>7 then

if days<14 then
```

```
f:=2;
end if;
else
f:=3;
end if;
update lib set fine=f where rno=100;
end;
Select * from lib;
```

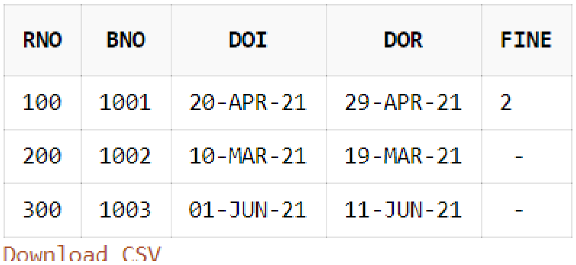
OUTPUT:

Before:



RNO	BNO	DOI	DOR	FINE
100	1001	20-APR-21	29-APR-21	-
200	1002	10-MAR-21	19-MAR-21	-
300	1003	01-JUN-21	11-JUN-21	-

After:



RNO	BNO	DOI	DOR	FINE
100	1001	20-APR-21	29-APR-21	2
200	1002	10-MAR-21	19-MAR-21	-
300	1003	01-JUN-21	11-JUN-21	-

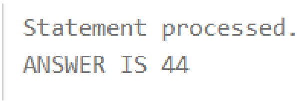
Download CSV

3. PL/SQL block that performs addition (+), subtraction (-), multiplication (*) and division (/) of two numbers as choice by the user.

STATEMENT:

```
declare
a number:=72;
b number:=28;
ans number(10);
choice char:='-';
begin
if choice='+' then
ans:=a+b;
elsif choice='-' then
ans:=a-b;
elsif choice='*' then
ans:=a*b;
elsif choice='/' then
ans:=a/b;
end if;
dbms_output.put_line('ANSWER IS ' || ans);
end;
```

OUTPUT:



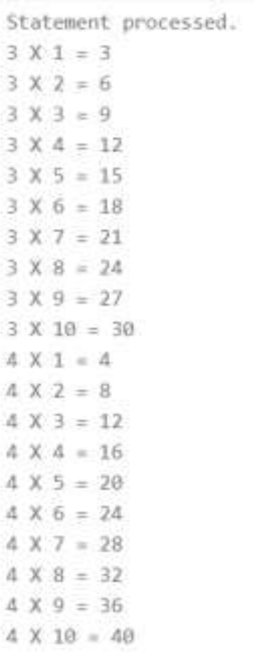
```
Statement processed.
ANSWER IS 44
```

4. PL/SQL block to generate multiplication table for 3 to n.

STATEMENT:

```
declare
str number:=3;
n number:=4;
begin
while str<=n loop
for i in 1..10 loop
dbms_output.put_line(str || ' X ' || i || ' = ' || (str*i));
end loop;
str:=str+1;
end loop;
end;
```

OUTPUT:



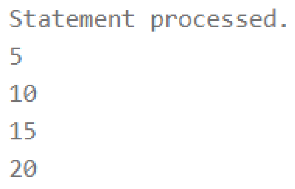
```
Statement processed.
3 X 1 = 3
3 X 2 = 6
3 X 3 = 9
3 X 4 = 12
3 X 5 = 15
3 X 6 = 18
3 X 7 = 21
3 X 8 = 24
3 X 9 = 27
3 X 10 = 30
4 X 1 = 4
4 X 2 = 8
4 X 3 = 12
4 X 4 = 16
4 X 5 = 20
4 X 6 = 24
4 X 7 = 28
4 X 8 = 32
4 X 9 = 36
4 X 10 = 40
```

5. PL/SQL block to print 5, 10, 15,20 by using For Loop

STATEMENT:

```
begin

for i in 5..20 loop

if mod(i,5)=0 then

dbms_output.put_line(i);

end if;

end loop;

end;
```

OUTPUT:



```
Statement processed.
5
10
15
20
```

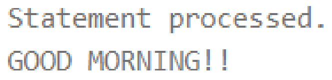
6. Pl/SQL block to display welcome message like good morning, good afternoon, good night depending on system time.

STATEMENT:

```
declare

curtime number(10);

begin

select to_char(sysdate,'HH24') into curtime from dual;

if curtime<12 then

dbms_output.put_line('GOOD MORNING!!');
```

```
elsif curtime>12 then
if curtime<17 then
dbms_output.put_line('GOOD AFTERNOON!!');
end if;
else
dbms_output.put_line('GOOD NIGHT!!');
end if;
end;
```

OUTPUT:



```
Statement processed.
GOOD MORNING!!
```

7. WAP that calculate simple interest for principal 1000, time 2 years and rate of interest varies from 5 to 15. Store it in a table.

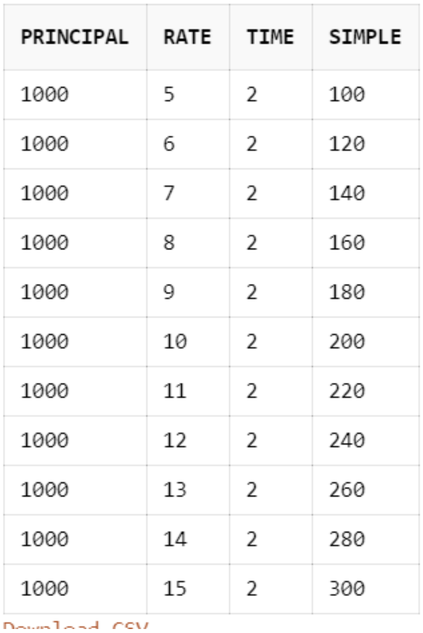
Principal time rate interest

STATEMENT:

```
create table pri(principal number(10),rate number(10),time number(10),simple float(5));
declare
p number:=1000;
t number:=2;
r number:=5;
si pri.simple%TYPE;
```

```
begin

for i in r..15 loop

si:=p*i*t/100;

insert into pri values(p,i,t,si);

end loop;

end;

select * from pri;
```

OUTPUT:



PRINCIPAL	RATE	TIME	SIMPLE
1000	5	2	100
1000	6	2	120
1000	7	2	140
1000	8	2	160
1000	9	2	180
1000	10	2	200
1000	11	2	220
1000	12	2	240
1000	13	2	260
1000	14	2	280
1000	15	2	300

Download CSV