

## Lab Assignment–9

### PL/SQL Lab Assignments

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;			

- 1) WAP to find the greatest of three numbers.

The screenshot shows the Oracle SQL Developer Live Worksheet interface. At the top, there's a toolbar with icons for Feedback, Help, and a user account. Below the toolbar, the title bar says "Live SQL". The main area is a "SQL Worksheet" containing the following PL/SQL code:

```
1 DECLARE
2   a INT:=34;
3   b INT:=78;
4   c INT:=12;
5 BEGIN
6   IF(a>b AND a>c)THEN
7     dbms_output.put_line('Greatest number is'||a);
8   ELSIF(b>a AND b>c)THEN
9     dbms_output.put_line('Greatest number is'||b);
10  ELSE
11    dbms_output.put_line('Greatest number is'||c);
12  END IF;
13 END;
14 
```

Below the code, the results of the execution are displayed:

```
Statement processed.
Greatest number is 78
```

2) 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

Live SQL

SQL Worksheet

1 DECLARE  
2 marks INT:=45; BEGIN  
3 IF(marks>80)THEN dbms\_output.put\_line('A grade'); ELSIF(marks>70)THEN  
4 dbms\_output.put\_line('B grade');  
5 ELSIF(marks>50)THEN  
6 dbms\_output.put\_line('C grade');  
7 ELSIF(marks>40)THEN  
8 dbms\_output.put\_line('D grade');  
9 ELSE  
10 dbms\_output.put\_line('E grade');  
11 END IF; END;  
12  
13

Statement processed.  
D grade

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

Live SQL

SQL Worksheet

1 DECLARE  
2 a INT:=5; k INT:=10; BEGIN  
3 dbms\_output.put\_line('Table of ||a|| is: ');  
4 FOR k IN 1..10 loop dbms\_output.put\_line(k\*a); END loop;  
5 END;  
6  
7

Statement processed.  
Table of 5 is:  
5  
10  
15  
20  
25  
30  
35  
40  
45  
50

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

The screenshot shows the Oracle Live SQL interface. In the SQL Worksheet, the following PL/SQL code is written:

```
1 DECLARE
2 fact INT:=1; k INT:=9; BEGIN
3 dbms_output.put_line('Factorial of'||k||' is: ');
4 WHILE k>0 loop fact:=fact*k; k:=k-1;
5 END loop; dbms_output.put_line(fact); END;
```

When run, the output window displays:

```
Statement processed.
Factorial of 9 is:
362880
```

5) WAP to find the reverse of a number(use exit when stement)

The screenshot shows the Oracle Live SQL interface. In the SQL Worksheet, the following PL/SQL code is written:

```
1 DECLARE
2 sums INT:=0; k INT:=124; BEGIN
3 dbms_output.put_line('Reverse of'||k||' is: ');
4 loop
5 exit WHEN k=0; sums:=(10*sums)+MOD(k,10); k:=k/10;
6 END loop; dbms_output.put_line(sums); END;
```

When run, the output window displays:

```
Statement processed.
Reverse of 124 is:
421
```

6) PL/SQL block to update total sal for empno 100 in Employee Table.

Table Employee: Eno,ename, bp,da,hra,total.

```

CREATE TABLE Employee_101915028(
    Eno INT,
    ename VARCHAR(20),
    bp INT,
    da INT,
    hra INT,
    total INT
);

INSERT INTO Employee_101915028 VALUES (100,'Brian',415,597,845,1020);
INSERT INTO Employee_101915028 VALUES (101,'Bill',451,927,835,1005);
INSERT INTO Employee_101915028 VALUES (102,'Ryan',454,957,85,10043);
INSERT INTO Employee_101915028 VALUES (103,'Gabe',415,9772,85,10230);
INSERT INTO Employee_101915028 VALUES (104,'Jim',435,975,835,12200);

SELECT*FROM Employee_101915028;

```

ENO	ENAME	BP	DA	HRA	TOTAL
100	Brian	415	597	845	1020
101	Bill	451	927	835	1005
102	Ryan	454	957	85	10043
103	Gabe	415	9772	85	10230
104	Jim	435	975	835	12200

[Download CSV](#)

```

DECLARE
newSal INT:=5000;
BEGIN
UPDATE Employee_101915028 SET total=newSal WHERE Eno=100;
END;

```

7) 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.

Live SQL

Feedback Help aagrawal\_be19@thapar.edu

SQL Worksheet

Clear Find Actions Save Run

```
1 drop table lib_101915028;
2
3 CREATE TABLE lib_101915028(
4 Rno INT PRIMARY KEY,
5 bookno INT,
6 doi DATE,
7 dor DATE,
8 fine INT
9 );
10
11 INSERT INTO lib_101915028 VALUES
12 (100,60,to_date('18/10/2021','dd/mm/yyyy'),to_date('16/11/2021','dd/mm/yyyy'),NULL);
13 INSERT INTO lib_101915028 VALUES
14 (101,65,to_date('2/10/2021','dd/mm/yyyy'),to_date('2/11/2021','dd/mm/yyyy'),NULL);
15 INSERT INTO lib_101915028
16 VALUES (102,30,to_date('8/10/2021','dd/mm/yyyy'),to_date('15/11/2021','dd/mm/yyyy'),NULL);
17 SELECT*FROM lib_101915028;
18
```

1 row(s) inserted.

RNO	BOOKNO	DOI	DOR	FINE
100	60	18-OCT-21	16-NOV-21	-
101	65	02-OCT-21	02-NOV-21	-
102	30	08-OCT-21	15-NOV-21	-

Download CSV  
3 rows selected.

Live SQL

Feedback Help aagrawal\_be19@thapar.edu

SQL Worksheet

Clear Find Actions Save Run

```
1
2
3 DECLARE
4 da INT; fine INT;
5 BEGIN
6 SELECT dor-doi INTO da FROM lib_101915028 WHERE Rno=100;
7 IF(da<7)THEN
8 fine:=1;
9 elsif(da>7 AND da<14)THEN fine:=2;
10 ELSE
11 fine:=3;
12 END IF;
13 dbms_output.put_line('Fine is'||fine);
14 END;
15
16
17
```

Statement processed.  
Fine is 3

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

The screenshot shows the Oracle SQL Developer Live SQL interface. The title bar says "Live SQL". The main area is titled "SQL Worksheet". It contains the following PL/SQL code:

```
1 DECLARE a INT; b INT; BEGIN
2 a:=&a;
3 b:=&b;
4 dbms_output.put_line('Addition is'||a+b);
5 dbms_output.put_line('Subtraction is'||a-b);
6 dbms_output.put_line('Multiplication is'||a*b);
7 dbms_output.put_line('Division is'||a/b);
8 END;
9
```

Below the code are standard toolbar buttons: Clear, Find, Actions, Save, and Run.

9) PI/SQL block to display welcome message like good morning, goodafternoon, good night depending on system time.

The screenshot shows the Oracle SQL Developer Live SQL interface. The title bar says "Live SQL". The main area is titled "SQL Worksheet". It contains the following PL/SQL code:

```
1 DECLARE
2 da INT;
3 BEGIN
4 SELECT to_char(CURRENT_TIMESTAMP, 'HH24') INTO da FROM dual;
5 IF(da>=3 AND da<12) THEN
6
7 dbms_output.put_line('Good Morning');
8 elsif(da>=12 AND da<18) THEN dbms_output.put_line('Good Afternoon');
9 ELSE
10 dbms_output.put_line('Good Night');
11 END IF;
12 END;
```

At the bottom of the interface, there is a message window showing the output of the executed statement:

```
Statement processed.
Good Morning
```