**15. a) Write an SQL code block these raise a user defined exception where business rule is voilated. BR for client\_ master table specifies when the value of bal\_due field is less than 0 handle the exception.**

SQL> create table client\_master(

2 id number(2) not null primary key,

3 bal\_due number(10,2)

4 );

SQL> insert into client\_master (id,bal\_due) values (1,500);

1 row created.

SQL> insert into client\_master (id,bal\_due) values (2,700);

1 row created.

SQL> insert into client\_master (id,bal\_due) values (3,-300);

1 row created.

SQL> insert into client\_master (id,bal\_due) values (4,-400);

1 row created.

SQL> insert into client\_master (id,bal\_due) values (5,800);

1 row created.

SQL> select \* from client\_master;

ID BAL\_DUE

---------- ----------

1 500

2 700

3 -300

4 -400

5 800

SQL> edit

Wrote file afiedt.buf

1 declare

2 c\_id client\_master.id%type := &c\_id;

3 c\_baldue client\_master.bal\_due%type;

4 balance\_exception exception;

5 begin

6 select bal\_due into c\_baldue from client\_master where id = c\_id;

7 if c\_baldue<0 then

8 raise balance\_exception;

9 else

10 dbms\_output.put\_line('Your balance due is: '||c\_baldue);

11 end if;

12 exception

13 when balance\_exception then

14 dbms\_output.put\_line('Your balance due is in negative: '||c\_baldue);

15\* end;

SQL> /

Enter value for c\_id: 2

old 2: c\_id client\_master.id%type := &c\_id;

new 2: c\_id client\_master.id%type := 2;

Your balance due is: 700

PL/SQL procedure successfully completed.

SQL> /

Enter value for c\_id: 3

old 2: c\_id client\_master.id%type := &c\_id;

new 2: c\_id client\_master.id%type := 3;

Your balance due is in negative: -300

PL/SQL procedure successfully completed.

SQL>