ASSIGNMENT: PL SQL

NAME: Gurunath Deshmukh

```
–table creation
create insert and procedure:
create table member(
mem_no varchar2(20),
mem_name varchar2(20) not null,
mem_type varchar2(20),
no_of_books number(4),
total_fine number(4),
constraint member primary key
(
mem_no
)
enable
);

create table book(
book_no varchar2(20) not null,
book_name varchar2(20),
author varchar2(20),
price varchar2(20),
no_of_books number(4),
constraint book primary key (book_no)
enable
);
```

```
create table trans(
book_no varchar2(20),
mem_no varchar2(20),
issue_date date,
due_date date,
return_date date,
foreign key (book_no) references book(book_no),
foreign key (mem_no) references member(mem_no)
);

–inserting values
insert into member values('100','sumeet','m','4',null);
insert into member values('200','devashish','y','10',null);
insert into member values('300','vaibhav','l','60',null);
insert into member values('400','gurunath','l','45',null);

insert into book values('1','rich dad poor dad','xx','399','30');
insert into book values('2','the alchemist','xx','199','20');
insert into book values('3','the atomic habbits','xx','399','50');
insert into book values('4','the heist','xx','599','35');

insert into trans values('1','100',sysdate,sysdate+7,null);
insert into trans values('2','200',sysdate,sysdate+7,null);
insert into trans values('3','300',sysdate,sysdate+7,null);
insert into trans values('4','400',sysdate,sysdate+7,null);
```

functions :

```
--function a

create or replace function book_valid(book_no1 book.book_no%type)
return boolean

is

 v_book number(10):=0;

begin

 select count(*) into v_book from book where book_no =book_no1;   if
v_book>0 then

 return true;

 else

 return false;

 end if;

end;


--function b

create or replace function mem_valid(mem_1 member.mem_no%type)
return boolean

is

 v_mem number(10):=0;

begin

 select count(*) into v_mem from member where mem_no =mem_1;   if
v_mem>0 then

 return true;

 else

 return false;

 end if;

end;
```

```
--function c

create or replace function borrow(mem_2 member.mem_no%type,book_2
book.book_no%type)  return boolean

is

 b_avail1 number(10):=0;

begin

 select count(*) into b_avail1 from trans where mem_no=mem_2 and
book_no=book_2 and return_date is null;   if b_avail1>0 then

 return true;

 else

 return false;

 end if;

end;


 --function e

 create or replace function limit(mem_3 member.mem_no%type)

 return boolean

 is

 l_exid number(10);

 m_limit number(10);


 check1 member.mem_type%type;

begin

 select count(*) into l_exid from trans where mem_no=mem_3 and
return_date is null;

 select mem_type into check1 from member where mem_no=mem_3 ;
```

```
 if(check1='m') then

 m_limit := 4;

 elsif(check1='y') then

 m_limit :=2;

 elsif(check1='l') then

 m_limit :=6;

 end if;


 if(l_exid>=m_limit) then

 return false;

 elsif(l_exid<m_limit) then

 return true;

 end if;

end;


 -- function f


create or replace function stock(book_no3 book.book_no%type)  return
boolean

is

 a_stock number(10);

begin

 select no_of_books into a_stock from book where book_no=book_no3;   if
a_stock>0 then

 return true;

 elsif a_stock<=0 then

 return false;

 end if;
```

```
end;
```

—procedure

```
set serveroutput on;

create or replace procedure issue(book_no book.book_no%type,mem_no
member.mem_no%type)  as

book_check boolean;

mem_check boolean;

b_avail boolean;

a exception;

b exception;

limit_1 boolean;

a_stock boolean;

day1 varchar2(20);


begin

--part1

dbms_output.put_line('part a');


begin

book_check :=book_valid(book_no);

 if(book_no is null)then

 raise a;

 end if;


 if book_check=true then
```

```
 dbms_output.put_line('book no is valid');   else
 dbms_output.put_line('book no is invalid');   end if;

 exception
 when a then
 dbms_output.put_line('error book no is null');   when no_data_found then
 dbms_output.put_line('all null');
end;

-- part 2
dbms_output.put_line('part b');

begin
 mem_check :=mem_valid(mem_no);
 if(mem_no is null) then
 raise b;
 end if;

 if mem_check=true then
 dbms_output.put_line('member no is valid');
 else
 dbms_output.put_line('member no is invalid');
 end if;

 exception
 when b then
 dbms_output.put_line('error member no is null');
```

```
 when no_data_found then
 dbms_output.put_line('all null');
end;

-- part 3
dbms_output.put_line('part c');

begin
b_avail :=borrow(mem_no,book_no);
 if(book_no is null or book_check=false )then
 raise a;
 end if;
 if(mem_no is null or mem_check=false) then
 raise b;
 end if;


 if b_avail=true then
 dbms_output.put_line('cannot borrow the book');   else
 dbms_output.put_line('can borrrow the book');
 end if;
 exception
 when a then
 dbms_output.put_line('error book no is null or put correct book_no');
 when b then
 dbms_output.put_line('error member no is null or put correct mem_no');
when no_data_found then
```

```
 dbms_output.put_line('all null');
end;

--part 5
begin
 dbms_output.put_line('part e');
 limit_1 :=limit(mem_no);
 if(mem_no is null or mem_check=false) then
 raise b;
 end if;


 if(limit_1=true) then
 dbms_output.put_line('you can borrow the book');
 elsif(limit_1=false) then
 dbms_output.put_line('you can not borrow the book');
 end if;

 exception
 when b then
 dbms_output.put_line('error member no is null or put correct mem_no');
end;

--part f
begin
 dbms_output.put_line('part f');
 a_stock :=stock(book_no);
```

```
 if(book_no is null or book_check=false )then

 raise a;

 end if;

 if(a_stock=true) then

 dbms_output.put_line('book stock is available');

 elsif(a_stock=false) then

 dbms_output.put_line('book stock is not available');

 end if;


 exception

 when a then

 dbms_output.put_line('error book no is null or put correct book_no');

end;


--part g

 dbms_output.put_line('part g');


begin

 if (book_check=true and mem_check=true and b_avail=false and
limit_1=true and a_stock=true) then   dbms_output.put_line('user can be
inserted');

 insert into trans values(book_no,mem_no,sysdate,sysdate+7,null);

 else

 dbms_output.put_line('user cannot be inserted');

 end if;

 end;
```

```
--part h;

begin

 dbms_output.put_line('part h');

 day1:=to_char(sysdate,'fmday');

 if(day1='saturday' or day1='sunday') then

 dbms_output.put_line('book cannot be issued on saturday and sunday');
else

 dbms_output.put_line('book can be issued on '|| to_char(sysdate,'day'));

 end if;



end;

end ;
```

to execute all :

```
execute issue(&book_no,&mem_no);
```

OUTPUT

```
part a
book no is valid
part b
member no is valid
part c
CANNOT BORROW THE BOOK
part e
you can borrow the book
part f
book stock is available
part G
USER CANNOT BE INSERTED
part H
BOOK CAN BE ISSUED ON  TUESDAY


PL/SQL procedure successfully completed.
```