ASSIGNMENT: PL SQL

NAME: VAIBHAV INGLE

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)

);

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);

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 ;
```

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;
```

TO EXECUTE ALL :

```
EXECUTE Issue(&BOOK_NO,&MEM_NO);
```

OUTPUT:

```
PL/SQL procedure successfully completed.

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.
```