

# PL/SQL CASE STUDY

**NAME: SUMEET KARBHARI**

## **Creation of Tables:**

### **Member:**

```
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  
);
```

### **Book:**

```
CREATE table book(  
  Book_no varchar2(20) NOT NULL,  
  Book_name varchar2(20),  
  Author varchar2(20),  
  price varchar2(20),  
  no_of_books number(4),  
  PRIMARY KEY (Book_no)  
);
```

### **Trans:**

```
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 Values:**

```
INSERT INTO MEMBER VALUES('100','SUMEET','M','4','150');
INSERT INTO MEMBER VALUES('200','DEVASHISH','Y','10','50');
INSERT INTO MEMBER VALUES('300','VAIBHAV','L','60','500');
INSERT INTO MEMBER VALUES('400','GURUNATH','L','45','250');
```

```
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:**

/\*part1\*/

```
CREATE OR REPLACE FUNCTION BOOK_VALID(BOOK_NO2 VARCHAR2)
return boolean
is book1 varchar2(20):=0;
```

begin

```
select count(*) into book1 from Book where Book_no=BOOK_NO2;
```

```
if(book1>0) then
return true;
```

```
else
return false;
end if;
```

end;

/

/\*part2\*/

```
CREATE OR REPLACE FUNCTION MEMBER_VALID(Mem_no2 VARCHAR2)
return boolean
is mem1 varchar2(20):=0;
```

```

begin
    select count(*) into mem1 from Member where Mem_no=Mem_no2;

    if(mem1>0) then
        return true;

    else
        return false;
    end if;
end;
/

```

/\*part3\*/

```

CREATE OR REPLACE FUNCTION Check_For_Book(Book_no2 VARCHAR2,Mem_no1
VARCHAR2)
return boolean
    is check_book varchar2(20):=0;
begin
    select count(*) into check_book from Trans where Book_no=Book_no2 and Mem_no =
Mem_no1 and return_date is null;
    if(check_book>0) then
        return true;

    else
        return false;
    end if;
end;
/

```

/\*part5\*/

```

CREATE OR REPLACE FUNCTION Handle_No_Of_Books(Mem_no1
Varchar2,membership varchar2)
return boolean
    is no_of_books number(4);

begin
    select No_of_books into no_of_books from member where Mem_no=Mem_no1;

```

```

    if(membership='M') then
        if(no_of_books>4) then
            return false;

        else
            return true;
        end if;

    ELSIF(membership='Y') then
        if(no_of_books>2) then
            return false;

        else
            return true;
        end if;

    ELSIF(membership='L') then
        if(no_of_books>6) then
            return false;

        else
            return true;
        end if;

    end if;

end;
/

/*part6*/
CREATE OR REPLACE FUNCTION Stocks(book_no1 varchar2)
    return boolean
    is count_books number(4);

begin

    SELECT no_of_books INTO count_books from book where book_no = book_no1;

    if(count_books>=1) then

```

```
        return true;

    else
        return false;

    end if;

end;
/
```

## **PROCEDURE:**

```
set serveroutput on;
create or replace procedure issue(book_no varchar2,Mem_no varchar2)
is
    check2 boolean;
    a exception;
    b exception;
    c1 boolean:=false;
    c2 boolean:=false;
    value13 varchar2(20);

    value12 varchar2(20):=book_no;

begin
begin
    check2:=BOOK_VALID(book_no);
    if(value12 is null) then
        raise b;

    elsif(check2=true) then
        DBMS_OUTPUT.PUT_LINE('Book No is Valid');

    elsif(check2 = false) then
        raise a;

    END IF;
exception

    when a then
```

```

        dbms_output.put_line('Book_no should be valid ');

    when b then

        dbms_output.put_line('NULL value');

End;

begin

check2 := MEMBER_VALID(Mem_no);
if(value12 is null) then
    raise b;

elseif(check2=true) then
    DBMS_OUTPUT.PUT_LINE('Member no is valid');

elseif(check2 = false) then
    raise a;

END IF;
exception

    when a then

        dbms_output.put_line('Member no must be valid');

    when b then

        dbms_output.put_line('NULL value');

End;

begin

    if(BOOK_VALID(book_no) = false or MEMBER_VALID(Mem_no)=false ) then

```

```

    raise b;
end if;

check2 := Check_For_Book(Book_no,Mem_no);


    if(check2=true) then
        raise a;

    else
        c2:=true;
        dbms_output.put_line('We can issue');

END IF;
exception

    when a then

        dbms_output.put_line('We cannot issue the book as you didnt submit the previous
book');

    when b then

        dbms_output.put_line('Invalid mem_no or book_no');

End;

begin
check2 := Handle_No_Of_Books(Mem_no,'M');
    if(check2=false) then
        raise a;

    else
        dbms_output.put_line('You are eligible for books');
    end if;
exception
    when a then
        dbms_output.put_line('please return the book first');

```

end;

```
begin
check2 := Stocks(book_no );
  if(check2=false) then
    raise a;

  else
    c1:=true;
    dbms_output.put_line('Books are Available in Stocks');
  end if;
exception
when a then
  dbms_output.put_line('Out of Stocks');
```

end;

```
begin

if(BOOK_VALID(book_no)=true and MEMBER_VALID(Mem_no) =true and c2=true and
c1=true and Stocks(book_no) =true) then
insert into trans values(Book_no,Mem_no,Sysdate,Sysdate+7,null);
  dbms_output.put_line('Its working');

else
  dbms_output.put_line('User cannont be entered ');
end if;
end;
```

```
begin

value13 := to_char(SYSDATE,'FMDAY');
if value13='SUNDAY'
  THEN
    dbms_output.put_line('We cannot issue the book on weekends');
```



```

ELSIF value13 ='SATURDAY' then
    dbms_output.put_line('We cannot issue the book on weekends');

ELSE
    dbms_output.put_line('We can issue the book');

END IF;

end;

end;
/

```

### **Execute Procedure:**

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

### **OUTPUT:**

**figure1**

```

PL/SQL procedure successfully completed.

Book No is Valid
Member no is valid
We cannot issue the book as you didnt submit the previous book
please return the book first
Books are Available in Stocks
User cannont be entered
We can issue the book

PL/SQL procedure successfully completed.

```