

## B.Sc. IV Sem

### DBMS Practical Programs

1. Create a database having two tables with the specified fields, to computerize a library system of a University College.

LibraryBooks (Accession number, Title, Author, Department, PurchaseDate, Price),

IssuedBooks (Accession number, Borrower) a) Identify primary and foreign keys.

a) Create the tables and insert at least 5 records in each table.

b) Delete the record of book titled "Database System Concepts".

c) Change the Department of the book titled "Discrete Maths" to "CS".

d) List all books that belong to "CS" department.

e) List all books that belong to "CS" department and are written by author "Navathe".

f) List all computer (Department="CS") that have been issued.

g) List all books which have a price less than 500 or purchased between "01/01/1999" and "01/01/2004".

LibraryBooks (Accession number, Title, Author, Department, PurchaseDate, Price)

```
SQL> create table LibraryBooks
 2 (
 3  accno number(4),
 4  title varchar(25),
 5  author varchar(10),
 6  dept varchar(5),
 7  purchasedate date,
 8  price number(4),
 9  constraint c1 primary key(accno)
10 );
```

Table created.

```
SQL> insert into LibraryBooks values
 2 (&accno,&'title',&'author',&'dept',&purchasedate',&price);
Enter value for accno: 1001
Enter value for title: DB System Concepts
Enter value for author: Korth
Enter value for dept: CS
Enter value for purchasedate: 1-Jan-2015
Enter value for price: 499
old 2: (&accno,&'title',&'author',&'dept',&purchasedate',&price)
```

```
new 2: (1001,'DB System Concepts','Korth','CS','1-Jan-2015',499)
```

```
1 row created.
```

```
SQL> /
```

```
Enter value for accno: 1002
```

```
Enter value for title: Database System
```

```
Enter value for author: Navathe
```

```
Enter value for dept: CS
```

```
Enter value for purchasedate: 1-Jan-1998
```

```
Enter value for price: 300
```

```
old 2: (&accno,&'&title','&author','&dept','&purchasedate',&price)
```

```
new 2: (1002,'Database System','Navathe','CS','1-Jan-1998',300)
```

```
1 row created.
```

```
SQL> /
```

```
Enter value for accno: 1003
```

```
Enter value for title: Discrete Maths
```

```
Enter value for author: SC Gupta
```

```
Enter value for dept: Maths
```

```
Enter value for purchasedate: 1-Nov-2003
```

```
Enter value for price: 510
```

```
old 2: (&accno,'&title','&author','&dept','&purchasedate',&price)
```

```
new 2: (1003,'Discrete Maths','SC Gupta','Maths','1-Nov-2003',510)
```

```
1 row created.
```

```
SQL> /
```

```
Enter value for accno: 1004
```

```
Enter value for title: M.Statistics
```

```
Enter value for author: SC Gupta
```

```
Enter value for dept: Stats
```

```
Enter value for purchasedate: 1-Apr-2021
```

```
Enter value for price: 999
```

```
old 2: (&accno,'&title','&author','&dept','&purchasedate',&price)
```

```
new 2: (1004,'M.Statistics','SC Gupta','Stats','1-Apr-2021',999)
```

```
1 row created.
```

```
SQL> /
```

```
Enter value for accno: 1005
```

```
Enter value for title: C
```

```
Enter value for author: Bala
```

```
Enter value for dept: CS
```

```
Enter value for purchasedate: 1-Jan-2020
```

Enter value for price: 700

old 2: (&accno,&title,&author,&dept,&purchasedate,&price)

new 2: (1005,'C','Bala','CS','1-Jan-2020',700)

1 row created.

SQL> select \* from Librarybooks;

ACCNO	TITLE	AUTHOR	DEPT	PURCHASED	PRICE
1001	DB System Concepts	Korth	CS	01-JAN-15	499
1002	Database System	Navathe	CS	01-JAN-98	300
1003	Discrete Maths	SC Gupta	Maths	01-NOV-03	510
1004	M.Statistics	SC Gupta	Stats	01-APR-21	999
1005	C	Bala	CS	01-JAN-20	700

Create IssuedBooks (Accession number, Borrower)

SQL> create table IssuedBooks

2 (

3 accno number(4),

4 borrower varchar(10),

5 constraint c2 foreign key(accno) references LibraryBooks(accno)

6 );

Table created.

SQL> insert into IssuedBooks values(&accno,&borrower');

Enter value for accno: 1001

Enter value for borrower: Rahul

old 1: insert into IssuedBooks values(&accno,&borrower')

new 1: insert into IssuedBooks values(1001,'Rahul')

1 row created.

SQL> /

Enter value for accno: 1005

Enter value for borrower: Suman

old 1: insert into IssuedBooks values(&accno,&borrower')

new 1: insert into IssuedBooks values(1005,'Suman')

1 row created.

SQL> select \* from IssuedBooks;

ACCNO	BORROWER
-------	----------

1001 Rahul  
1005 Suman

b) Delete the record of book titled "Database System Concepts".

```
SQL> Delete from LibraryBooks  
2  where title='Database System Concepts';
```

0 rows deleted.

c) Change the Department of the book titled "Discrete Maths" to "CS".

```
SQL> update LibraryBooks  
2  set dept='CS'  
3  where title='Discrete Maths';
```

1 row updated.

d) List all books that belong to "CS" department.

```
SQL> select * from LibraryBooks  
2  where dept='CS';
```

ACCNO	TITLE	AUTHOR	DEPT	PURCHASED	PRICE
1001	DB System Concepts	Korth	CS	01-JAN-15	499
1002	Database System	Navathe	CS	01-JAN-98	300
1003	Discrete Maths	SC Gupta	CS	01-NOV-03	510
1005	C	Bala	CS	01-JAN-20	700

e) List all books that belong to "CS" department and are written by author "Navathe".

```
SQL> select * from LibraryBooks  
2  where dept='CS' and author='Navathe';
```

ACCNO	TITLE	AUTHOR	DEPT	PURCHASED	PRICE
1002	Database System	Navathe	CS	01-JAN-98	300

f) List all computer books (Department="CS") that have been issued.

```
SQL> select * from LibraryBooks  
2  where dept='CS' and accno in (select accno from IssuedBooks);
```

ACCNO	TITLE	AUTHOR	DEPT	PURCHASED	PRICE
-------	-------	--------	------	-----------	-------

```
-----
1001 DB System Concepts Korth CS 01-JAN-15 499
1005 C Bala CS 01-JAN-20 700
```

g) List all books which have a price less than 500 or purchased between "01/01/1999" and "01/01/2004"

```
SQL> select * from LibraryBooks
2 where price<500 or purchasedate between '01-Jan-1999' and '01-Jan-2004';
```

```
-----
ACCNO TITLE AUTHOR DEPT PURCHASED PRICE
-----
1001 DB System Concepts Korth CS 01-JAN-15 499
1002 Database System Navathe CS 01-JAN-98 300
1003 Discrete Maths SC Gupta CS 01-NOV-03 510
```

```
SQL> commit;
```

Commit complete.

**2. Create a database having three tables to store the details of students of Computer Department in your college.**

**1. Personal information about Student (College roll number, Name of student, Date of birth, Address, Marks(rounded off to whole number) in percentage at 10 + 2, Phone number). College Roll Number is the primary key**

**2. Paper Details (Paper code, Name of the Paper). Paper Code is the Primary Key**

**3. Student's Academic and Attendance details (College roll number, Paper Code, Attendance, Marks in home examination). College Roll No is a Foreign key & PaperCode is a Foreign Key.**

```
create table StudentInformation
(
RollNo number(12),
name varchar(10),
dob date,
address varchar(15),
marks_plus2 number(3,0),
phno number(10),
constraint c3 primary key(rollno)
);
```

```
create table PaperDetails
(
```

```
papercode varchar(6),
papername varchar(15),
constraint c4 primary key(papercode)
);
```

```
create table AcadAttendance
(
RollNo number(12),
papercode varchar(6),
attendance number(5,2),
marks number(5,2),
constraint c5 foreign key(RollNo) references StudentInformation(RollNo),
constraint c6 foreign key(papercode) references PaperDetails(papercode)
);
```

College Roll Number is a foreign key and paper code is also a foreign key.

a) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.

```
SQL> insert into StudentInformation
values(&RollNo,&name','&dob','&address',&marks_plus2,&phno);
Enter value for rollno: 5001
Enter value for name: ABC
Enter value for dob: 1-Jan-2000
Enter value for address: Hyderabad
Enter value for marks_plus2: 76
Enter value for phno: 9988776655
old 1: insert into StudentInformation
values(&RollNo,&name','&dob','&address',&marks_plus2,&phno)
new 1: insert into StudentInformation values(5001,'ABC','1-Jan-
2000','Hyderabad',76,9988776655)
```

1 row created.

```
SQL> /
Enter value for rollno: 5002
Enter value for name: DEF
Enter value for dob: 2-Feb-2001
Enter value for address: Warangal
Enter value for marks_plus2: 74
Enter value for phno: 8877665544
old 1: insert into StudentInformation
values(&RollNo,&name','&dob','&address',&marks_plus2,&phno)
new 1: insert into StudentInformation values(5002,'DEF','2-Feb-
2001','Warangal',74,8877665544)
```

1 row created.

SQL> /

Enter value for rollno: 5003

Enter value for name: xyz

Enter value for dob: 4-Mar-2002

Enter value for address: Khammam

Enter value for marks\_plus2: 75

Enter value for phno: 7894561234

old 1: insert into StudentInformation

values(&RollNo,&name','&dob','&address',&marks\_plus2,&phno)

new 1: insert into StudentInformation values(5003,'xyz','4-Mar-2002','Khammam',75,7894561234)

1 row created.

SQL> /

Enter value for rollno: 5004

Enter value for name: pqr

Enter value for dob: 5-Apr-2001

Enter value for address: Kurnool

Enter value for marks\_plus2: 54

Enter value for phno: 8965478654

old 1: insert into StudentInformation

values(&RollNo,&name','&dob','&address',&marks\_plus2,&phno)

new 1: insert into StudentInformation values(5004,'pqr','5-Apr-2001','Kurnool',54,8965478654)

1 row created.

SQL> /

Enter value for rollno: 5005

Enter value for name: rahul

Enter value for dob: 6-Jun-2001

Enter value for address: Anantapur

Enter value for marks\_plus2: 87

Enter value for phno: 8965478965

old 1: insert into StudentInformation

values(&RollNo,&name','&dob','&address',&marks\_plus2,&phno)

new 1: insert into StudentInformation values(5005,'rahul','6-Jun-2001','Anantapur',87,8965478965)

1 row created.

SQL> insert into PaperDetails values('&papercode','&papername');

Enter value for papercode: Paper1

Enter value for papername: Prog with C

```
old 1: insert into PaperDetails values('&papercode','&papername')
new 1: insert into PaperDetails values('Paper1','Prog with C')
```

1 row created.

```
SQL> /
```

```
Enter value for papercode: Paper2
```

```
Enter value for papername: Prog with C++
```

```
old 1: insert into PaperDetails values('&papercode','&papername')
```

```
new 1: insert into PaperDetails values('Paper2','Prog with C++')
```

1 row created.

```
SQL> /
```

```
Enter value for papercode: Paper3
```

```
Enter value for papername: DataStructures
```

```
old 1: insert into PaperDetails values('&papercode','&papername')
```

```
new 1: insert into PaperDetails values('Paper3','DataStructures')
```

1 row created.

```
SQL> /
```

```
Enter value for papercode: Paper4
```

```
Enter value for papername: DBMS
```

```
old 1: insert into PaperDetails values('&papercode','&papername')
```

```
new 1: insert into PaperDetails values('Paper4','DBMS')
```

1 row created.

```
SQL> /
```

```
Enter value for papercode: Paper5
```

```
Enter value for papername: Prog with Java
```

```
old 1: insert into PaperDetails values('&papercode','&papername')
```

```
new 1: insert into PaperDetails values('Paper5','Prog with Java')
```

1 row created.

```
SQL> select * from PaperDetails;
```

```
PAPERC PAPERNAME
```

```
-----
```

```
Paper1 Prog with C
```

```
Paper2 Prog with C++
```

```
Paper3 DataStructures
```

```
Paper4 DBMS
```

```
Paper5 Prog with Java
```



SQL> /

Enter value for rollno: 5001

Enter value for papercode: Paper1

Enter value for attendance: 60

Enter value for marks: 80

old 1: insert into AcadAttendance values(&RollNo,'&PaperCode',&attendance,&marks)

new 1: insert into AcadAttendance values(5001,'Paper1',60,80)

1 row created.

SQL> /

Enter value for rollno: 5002

Enter value for papercode: Paper2

Enter value for attendance: 78

Enter value for marks: 90

old 1: insert into AcadAttendance values(&RollNo,'&PaperCode',&attendance,&marks)

new 1: insert into AcadAttendance values(5002,'Paper2',78,90)

1 row created.

b) Design a query that will return the records (from the second table) along with the name of student from the first table, related to students who have more than 75% attendance and more than 60% marks in paper2.

```
select unique s.name, a.papercode, a.papername
from StudentInformation s, Paperdetails p, AcadAttendance a
where a.attendance>75 and a.marks>60 and a.papercode='paper2' and s.rollno=a.rollno;
```

```
select unique s.name, a.papercode
from StudentInformation s, Paperdetails p, AcadAttendance a
where a.attendance>75 and a.marks>60 and a.papercode='paper2' and s.rollno=a.rollno;
```

c) List all students who live in "Warangal" and have marks greater than 60 in paper1.

```
select unique s.name
from StudentInformation s, AcadAttendance a
where s.address='Warangal' and a.marks>60 and a.papercode='paper1';
```

d) Find the total attendance and total marks obtained by each student.

```
select sum(attendance), sum(marks)
from AcadAttendance
group by rollno;
```

e) List the name of student who has got the highest marks in paper2

```
select name, rollno
from StudentInformation
where rollno=(select rollno from AcadAttendance
              where papercode='paper2' and marks=(select max(marks) from AcadAttendance
              where papercode='paper2'));
```

```
select rollno
from AcadAttendance
where papercode='paper2' and marks=(select max(marks) from AcadAttendance where
papercode='paper2');
```

**3. Create the following tables and answer the queries given below:**

**Customer (CustID, email, Name, Phone, ReferrerID)**

**Bicycle (BicycleID, DatePurchased, Color, CustID, ModelNo)**

**BicycleModel(ModelNo, Manufacturer, Style)**

**Service (StartDate, BicycleID, EndDate)**

- 
- (i). In Customer table, CustID is the primary key & ReferrerId is a foreign key
  - (ii). In Bicycle table, BicycleID is the primary key , CustId is a foreign key & ModelNo is also foreign key
  - (iii) In BicycleModel table, ModelNo is the primary key.
  - (iv) In Service table, BicycleId is a foreign key.

Create table in this order: (i), (iii), (ii), (iv)

a) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.

-----

(i) Customer (CustID, email, Name, Phone, ReferrerID)

In Customer table, CustID is the primary key & ReferrerId is a foreign key

Create table Customer

```
(
CustID varchar(4),
email varchar(20),
name varchar(15),
phone number(10),
ReferrerId varchar(4),
constraint c31 primary key(CustId)
```

);

insert into Customer values('&CustId','&email','&name','&phone','&ReferrerId');

SQL> insert into Customer values('&CustId','&email','&name','&phone','&ReferrerId');

Enter value for custid: C1

Enter value for email: abc@gmail.com

Enter value for name: ABC

Enter value for phone: 8899889988

Enter value for referrerid: NA

old 1: insert into Customer values('&CustId','&email','&name','&phone','&ReferrerId')

new 1: insert into Customer values('C1','abc@gmail.com','ABC',8899889988,'NA')

1 row created.

SQL> /

Enter value for custid: C2

Enter value for email: def@gmail.com

Enter value for name: DEF

Enter value for phone: 7788990099

Enter value for referrerid: C1

old 1: insert into Customer values('&CustId','&email','&name','&phone','&ReferrerId')

new 1: insert into Customer values('C2','def@gmail.com','DEF',7788990099,'C1')

1 row created.

SQL> select \* from Customer;

CUST	EMAIL	NAME	PHONE	REFE
C1	abc@gmail.com	ABC	8899889988	NA
C2	def@gmail.com	DEF	7788990099	C1

(ii) BicycleModel(ModelNo, Manufacturer, Style)

In BicycleModel table, ModelNo is the primary key.

Create table BicycleModel

(

ModelNo number(4),

Manufacturer varchar(10),

Style varchar(10),

constraint c32 primary key(ModelNo)

);

```
insert into BicycleModel values(&ModelNo,'&Manufacturer','&style');
```

```
SQL> insert into BicycleModel values(&ModelNo,'&Manufacturer','&style');  
Enter value for modelno: 2020  
Enter value for manufacturer: Honda  
Enter value for style: RoadBike  
old 1: insert into BicycleModel values(&ModelNo,'&Manufacturer','&style')  
new 1: insert into BicycleModel values(2020,'Honda','RoadBike')
```

1 row created.

```
SQL> /  
Enter value for modelno: 2021  
Enter value for manufacturer: Hero  
Enter value for style: ElectrBike  
old 1: insert into BicycleModel values(&ModelNo,'&Manufacturer','&style')  
new 1: insert into BicycleModel values(2021,'Hero','ElectrBike')
```

1 row created.

-----  
(iii) Bicycle (BicycleID, DatePurchased, Color, CustID, ModelNo)

In Bicycle table, BicycleID is the primary key , CustID is a foreign key & ModelNo is also foreign key

```
Create table Bicycle  
(  
  BicycleId number(2),  
  DatePurchased date,  
  color varchar(8),  
  CustId varchar(4),  
  ModelNo number(4),  
  constraint c33 primary key(BicycleId),  
  constraint c34 foreign key(CustId) references Customer(CustID),  
  constraint c35 foreign key(ModelNo) references BicycleModel(ModelNo)  
);
```

```
insert into Bicycle values(&BicycleId,'&DataPurchased','&color','&CustId',&ModelNo);
```

```
SQL> insert into Bicycle values(&BicycleId,'&DataPurchased','&color','&CustId',&ModelNo);  
Enter value for bicycleid: 21  
Enter value for datapurchased: 4-jun-2021  
Enter value for color: red  
Enter value for custid: C1
```

Enter value for modelno: 2021

old 1: insert into Bicycle values(&BicycleId,&DataPurchased,&color,&CustId,&ModelNo)

new 1: insert into Bicycle values(21,'4-jun-2021','red','C1',2021)

1 row created.

SQL> /

Enter value for bicycleid: 20

Enter value for datapurchased: 6-dec-2020

Enter value for color: black

Enter value for custid: C2

Enter value for modelno: 2020

old 1: insert into Bicycle values(&BicycleId,&DataPurchased,&color,&CustId,&ModelNo)

new 1: insert into Bicycle values(20,'6-dec-2020','black','C2',2020)

1 row created.

SQL> select \* from Bicycle;

BICYCLEID	DATEPURCH	COLOR	CUST	MODELNO
21	04-JUN-21	red	C1	2021
20	06-DEC-20	black	C2	2020

(iv)

Service (StartDate, BicycleID, EndDate)

In Service table, Bicycleid is a foreign key.

Create table Service

```
(
  startdate date,
  Bicycleid number(2),
  enddate date,
  constraint c36 foreign key(Bicycleid) references Bicycle(Bicycleid)
);
```

insert into Service values('&startdate',&BicycleId,&enddate');

SQL> insert into Service values('&startdate',&BicycleId,&enddate');

Enter value for startdate: 15-jun-2021

Enter value for bicycleid: 21

Enter value for enddate: 30-jun-2021

old 1: insert into Service values('&startdate',&BicycleId,&enddate')

new 1: insert into Service values('15-jun-2021',21,'30-jun-2021')

1 row created.

SQL> /

Enter value for startdate: 1-jan-2021

Enter value for bicycleid: 20

Enter value for enddate: 15-jan-2021

old 1: insert into Service values('&startdate',&BicycleId,&enddate')

new 1: insert into Service values('1-jan-2021',20,'15-jan-2021')

1 row created.

SQL> select \* from Bicycle;

BICYCLEID	DATEPURCH	COLOR	CUST	MODELNO
21	04-JUN-21	red	C1	2021
20	06-DEC-20	black	C2	2020

Data from 4 tables:

SQL> select \* from Customer;

CUST	EMAIL	NAME	PHONE	REFE
C1	abc@gmail.com	ABC	8899889988	NA
C2	def@gmail.com	DEF	7788990099	C1

SQL> select \* from BicycleModel;

MODELNO	MANUFACTUR	STYLE
2020	Honda	RoadBike
2021	Hero	ElectrBike

SQL> select \* from Bicycle;

BICYCLEID	DATEPURCH	COLOR	CUST	MODELNO
21	04-JUN-21	red	C1	2021
20	06-DEC-20	black	C2	2020

SQL> select \* from service;

STARTDATE	BICYCLEID	ENDDATE
15-JUN-21	21	30-JUN-21

01-JAN-21 20 15-JAN-21

---

b) List all the customers who have the bicycles manufactured by manufacturer "Honda".

names of the customers - Customer table

Manufacturer - BicycleModel table

```
select modelno
from BicycleModel
where manufacturer='Honda'; // Find modelno for the manufacturer Honda
```

```
select custid
from Bicycle
where modelno=(select modelno from BicycleModel where manufacturer='Honda'); //
find custid for Honda
```

```
select name
from Customer
where custid=(select custid from Bicycle where modelno=(select modelno from
BicycleModel where manufacturer='Honda'));
```

c) List the bicycles purchased by the customers who have been referred by Customer "C1".

Customer table

Bicycle table

```
select Custid from Customer
where ReferrerID='C1'; // It finds Custid for the referrer C1.
```

```
select * from Bicycle
where custid=(select Custid from Customer where ReferrerID='C1');
```

```
SQL> select * from Bicycle
2 where custid=(select Custid from Customer where ReferrerID='C1');
```

```
BICYCLEID DATEPURCH COLOR CUST MODELNO
-----
20 06-DEC-20 black C2 2020
```

d) List the manufacturer of red colored bicycles.

Bicycle (BicycleID, DatePurchased, Color, CustID, ModelNo)

BicycleModel(ModelNo, Manufacturer, Style)

```
select ModelNo
from Bicycle           // Finds the modelnos which are red color
where color='red';
```

```
MODELNO
-----
2021
```

```
select manufacturer
from BicycleModel
where ModelNo=(select ModelNo from Bicycle where color='red');
```

```
MANUFACTUR
-----
Hero
```

e) List the models of the bicycles given for service.

Bicycle (BicycleID, DatePurchased, Color, CustID, ModelNo)

Service (StartDate, BicycleID, EndDate)

```
select BicycleId
from Service;
```

```
BICYCLEID
-----
21
20
```

```
select ModelNo
from Bicycle
where BicycleId = (select BicycleId from Service); //Not suggested for multiple-rows
```



```
select ModelNo
from Bicycle
where BicycleId in (select BicycleId from Service); // when multiple-rows exist, then make
use of "in".
```

#### 4. Write a PL/SQL Program to demonstrate Procedure.

Create or replace procedure addvalues(a in number,b in number, c out number)

is

Begin

c:=a+b;

end;

/

-----

declare

x number;

y number;

z number;

begin

x:=&x;


y:=&y;

addvalues(x,y,z);

dbms\_output.put\_line('Sum of Two number='||z);

end;

/

 Oracle SQL\*Plus

File Edit Search Options Help

SQL> set serveroutput on

SQL> Create or replace procedure addvalues(a in number,b in number, c out number)

2 is

3 Begin

4 c:=a+b;

5 end;

6 /

Procedure created.

SQL> declare

2 x number;

3 y number;

4 z number;

5 begin

6 x:=&x;

7 y:=&y;

8 addvalues(x,y,z);

9 dbms\_output.put\_line('Sum of Two number='||z);

10 end;

11 /

Enter value for x: 20

old 6: x:=&x;

new 6: x:=20;

Enter value for y: 35

old 7: y:=&y;

new 7: y:=35;

Sum of Two number=55

PL/SQL procedure successfully completed.

SQL> |

## 5. Write a PL/SQL Program to demonstrate Function.

```
Create or replace function addnum(a number,b number)
return number
is
c number;
Begin
  c:=a+b;
  return(c);
end;
/
```

```
-----
declare
x number;
y number;
z number;
begin
  x:=&x;
  y:=&y;
  z:=addnum(x,y);
  dbms_output.put_line('Sum of Two number='||z);
end;
/
```

### Output:

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> set serveroutput on
SQL> Create or replace function addnum(a number,b number)
 2  return number
 3  is
 4  c number;
 5  Begin
 6  c:=a+b;
 7  return(c);
 8  end;
 9  /

Function created.

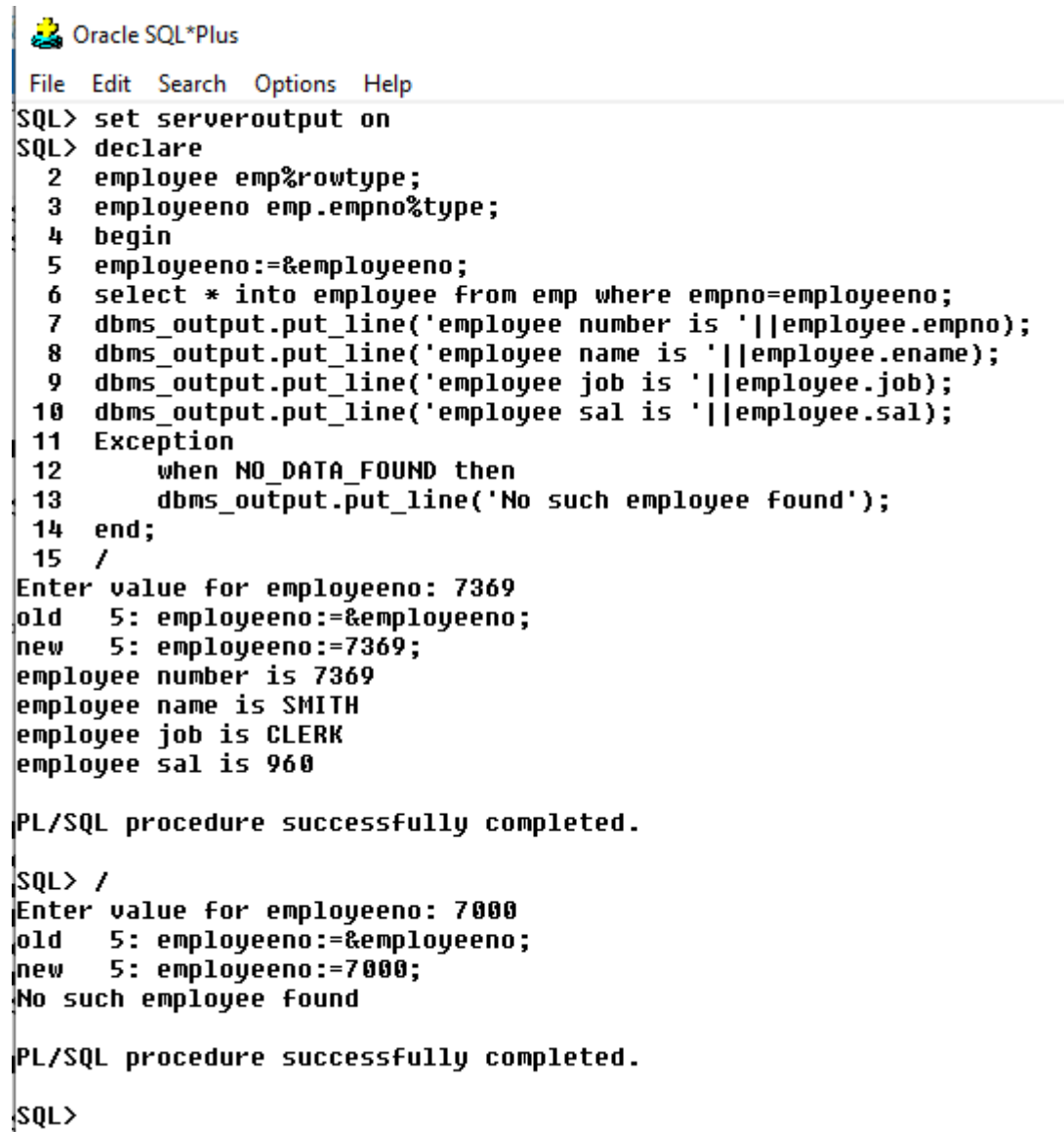
SQL> declare
 2  x number;
 3  y number;
 4  z number;
 5  begin
 6  x:=&x;
 7  y:=&y;
 8  z:=addnum(x,y);
 9  dbms_output.put_line('Sum of Two number='||z);
10 end;
11 /
Enter value for x: 50
old 6: x:=&x;
new 6: x:=50;
Enter value for y: 75
old 7: y:=&y;
new 7: y:=75;
Sum of Two number=125

PL/SQL procedure successfully completed.

SQL> |
```

## 6. Write a PL/SQL program to Handle Exceptions.

```
declare
employee emp%rowtype;
employeeno emp.empno%type;
begin
employeeno:=&employeeno;
select * into employee from emp where empno=employeeno;
dbms_output.put_line('employee number is '||employee.empno);
dbms_output.put_line('employee name is '||employee.ename);
dbms_output.put_line('employee job is '||employee.job);
dbms_output.put_line('employee sal is '||employee.sal);
Exception
  when NO_DATA_FOUND then
    dbms_output.put_line('No such employee found');
end;
/
```



```
Oracle SQL*Plus
File Edit Search Options Help
SQL> set serveroutput on
SQL> declare
  2  employee emp%rowtype;
  3  employeeno emp.empno%type;
  4  begin
  5  employeeno:=&employeeno;
  6  select * into employee from emp where empno=employeeno;
  7  dbms_output.put_line('employee number is '||employee.empno);
  8  dbms_output.put_line('employee name is '||employee.ename);
  9  dbms_output.put_line('employee job is '||employee.job);
 10  dbms_output.put_line('employee sal is '||employee.sal);
 11  Exception
 12      when NO_DATA_FOUND then
 13          dbms_output.put_line('No such employee found');
 14  end;
 15  /
Enter value for employeeno: 7369
old  5: employeeno:=&employeeno;
new  5: employeeno:=7369;
employee number is 7369
employee name is SMITH
employee job is CLERK
employee sal is 960

PL/SQL procedure successfully completed.

SQL> /
Enter value for employeeno: 7000
old  5: employeeno:=&employeeno;
new  5: employeeno:=7000;
No such employee found

PL/SQL procedure successfully completed.

SQL>
```