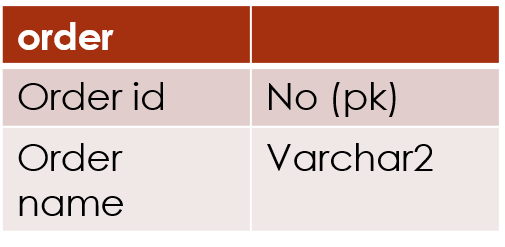
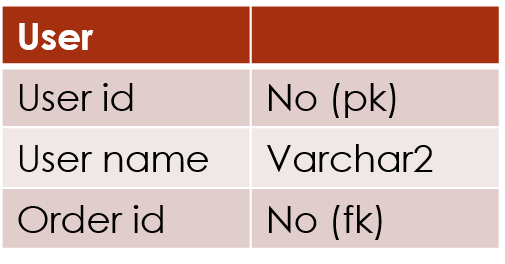
**Lab 1**

**INTRODUCTION TO DENORMALIZATION**

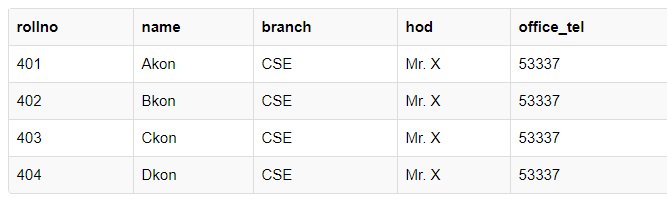
**Task:1:**De-normalize following table:

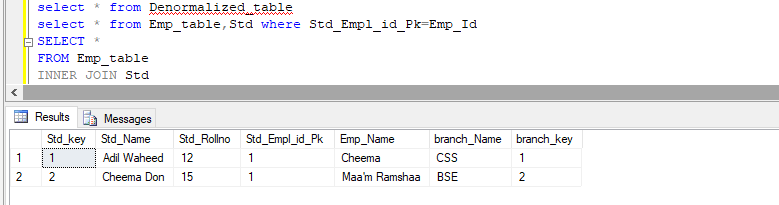


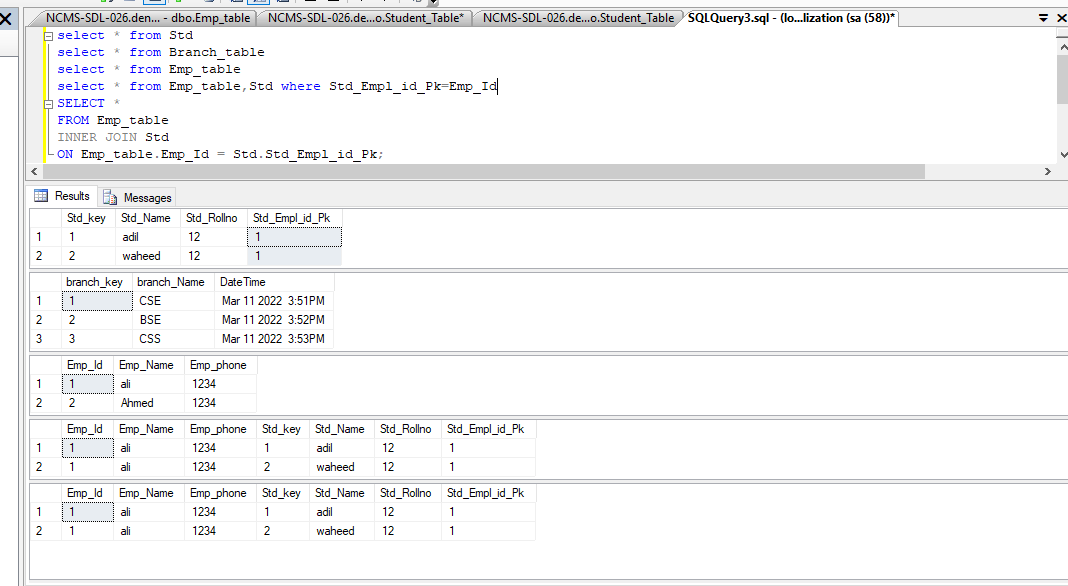
**Solution:**

|  |  |
| --- | --- |
| **User** |  |
| User id | No(pk) |
| UserName | Varchar2 |
| Order id | No |
| Order Name | Varchar2 |

**Task:2:**Normalized below table and then de-normalized it if required

* **Solution:** **[DENORMALIZED]:**



* **NORMALIZED**

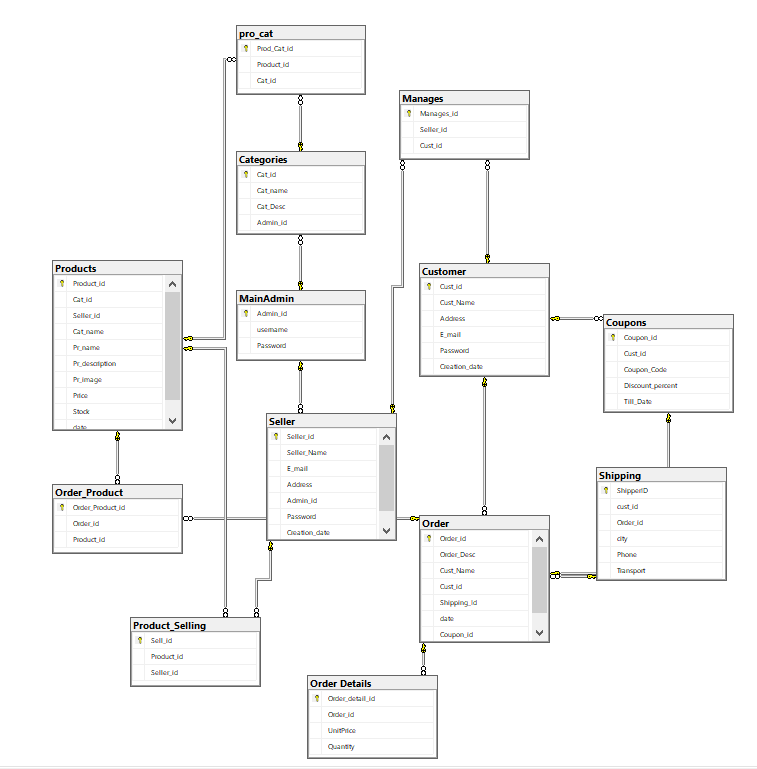
**Task 3:**Download any database and apply following operation

1. Check the database and tables architecture
2. Normalize if any table is not normalize
3. Check for table who need to be denormalized.

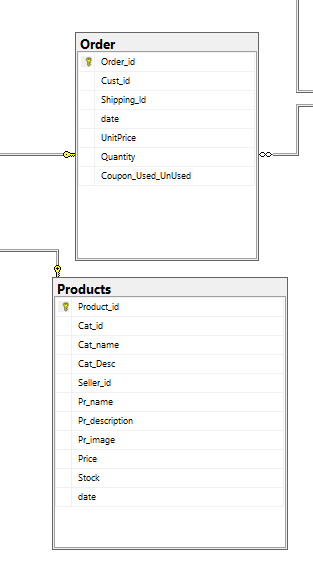
**Solution:**

**This Database Already Normalized**

**We Need To denormalized Those Table like Order and Order Details,Product and Category**

**Normalized Database:**

**Normalized Table Of Order And Product:**



**Lab 2**

**ORDBMS –CRUD OPERATION & Object Creation**

**Task1**:Apply CRUD operation on following table student(name, address, phone no , father name ),teacher(teacher name ,subject, semester ,etc) suggest datatype by your self.

Code:

**create table Student (**

**name varchar2(4000),**

**address varchar2(4000),**

**phone\_No number,**

**father\_name varchar2(4000),**

**std\_id number not null constraint student\_pk primary key**

**);**

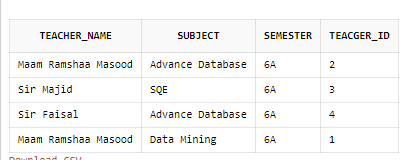
**/**

**create table Teacher (**

**teacher\_name varchar2(40),**

**subject varchar2(40),**

**semester varchar2(40),**

 **teacger\_id number not null constraint teacher\_pk primary key**

**);**

**/**

**select \* from Teacher;**

**select \* from Student;**

**INSERT INTO Teacher VALUES**

**(**

**-- 'Maam Ramshaa Masood','Data Mining','6A',1**

**-- 'Maam Ramshaa Masood','Advance Database','6A',2**

**-- 'Sir Majid','SQE','6A',3**

**'Sir Faisal','Advance Database','6A',4**

**);**

**INSERT INTO Student VALUES**

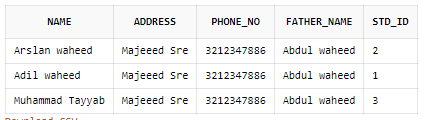
**(**

**'Adil waheed','Majeeed Sre',03212347886,'Abdul waheed',1**

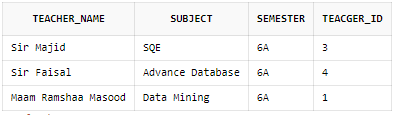
**'Arslan waheed','Majeeed Sre',03212347886,'Abdul waheed',2**

**'Muhammad Tayyab','Majeeed Sre',03212347886,'Abdul waheed',3**

**);**

****

**DELETE FROM Teacher t WHERE t.teacger\_id =2;**

****

**Task2:**create employee(employee no,employee name and company object details) table using department object. Department object holds dept no, dept name, dept strength etc suggest data type by your self.

**Code:**

**create or replace type em\_obj as object**

**( empname varchar2(20) ,empdes varchar2(5),phone number**

**);**

**create table tbemployee**

**( empno number(4), emp\_details em\_obj**

**);**

**select \* from tbemployee;**

**INSERT INTO tbemployee VALUES**

**(**

**--45,em\_obj('ali','AM',12345)**

**-- 49,em\_obj('Adil','AM',12345)**

**-- 46,em\_obj('Farook','AM',5643)**

**-- 47,em\_obj('Sohail','FM',12345)**

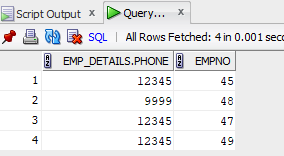
**-- 48,em\_obj('Cheema','DON',9999)**

**);**

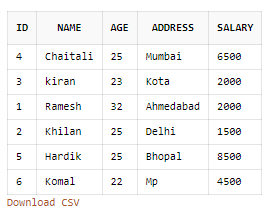
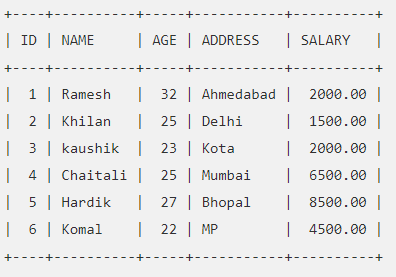
**select t.emp\_details.phone,t.empno from tbemployee t;**

**DELETE FROM tbemployee t WHERE t.emp\_details.phone =5643;**

**OUTPUT:**



**Task3:**Create customer table and perform all crud operation

 **Code**:

**create table Customer (**

**id number constraint customer\_pk primary key,**

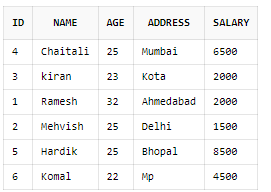
**name varchar2(4000),**

**age number,**

**address varchar2(4000),**

**salary varchar2(4000)**

**);**

**INSERT INTO Customer VALUES**

**(**

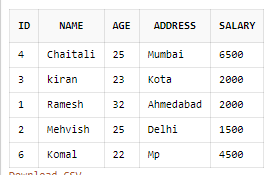
**1,'Ramesh',32,'Ahmedabad','2000'**

**2,'Khilan',25,'Delhi','1500'**

**3,'kiran',23,'Kota','2000'**

**4,'Chaitali',25,'Mumbai','6500'**

**5,'Hardik',25,'Bhopal','8500'**

 **6,'Komal',22,'Mp','4500'**

**);**

**/**

**select \* from Customer;**

**UPDATE Customer SET name = 'Mehvish' WHERE id=2;**

**DELETE FROM Customer t WHERE t.id ='5';**

**Task4:**Create simple calculator using variables

**accept x integer prompt 'please Enter Number'**

**accept y integer prompt 'please Enter Number2'**

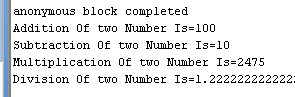
**declare**

**SUBTYPE number1 IS integer;**

**--num number1;**

**--num2 number1;**

**num number1 := '$x';**

**num2 number1 := '$y';**

**begin**

**--num := '$x';**

**--num2 := '$y';**

**dbms\_output.put\_line('Addition Of two Number Is=' || (num+num2));**

**dbms\_output.put\_line('Subtraction Of two Number Is=' || (num-num2));**

**dbms\_output.put\_line('Multiplication Of two Number Is=' || (num\*num2));**

**dbms\_output.put\_line('Division Of two Number Is=' || (num/num2));**

**end;/**

**Lab3**

**ORDBMS –Method Calling**

**ORDBMS METHOD CALLING**

**Task 1**:Develop Simple Calculator using function

**create or replace function Addition(num1 in number,num2 in number)**

**return varchar2 is**

**begin return ('Addition:'||(num1+num2));**

**end;**

**create or replace function Sub(num1 in number,num2 in number)**

**return varchar2 is**

**begin return ('Subtraction:'||(num1-num2));**

**end;**

**create or replace function Mul(num1 in number,num2 in number)**

**return varchar2 is**

**begin return ('Multiplication:'||(num1\*num2));**

**end;**

**create or replace function Div(num1 in number,num2 in number)**

**return varchar2 is**

**begin return ('Division:'||(num1/num2));**

**end;**

**/**

**--by using query**

**Select welcomes('ali') From dual;**

**set serveroutput on;**

**Declare**

**a char(10);**

**r1 number;**

**begin**

**a:='/';**

**if a='+' then**

**dbms\_output.put\_line(Addition(4,4));**

**elsif a='-' then**

**dbms\_output.put\_line(Sub(4,4));**

**elsif a='\*' then**

**dbms\_output.put\_line(Mul(4,4));**

**elsif a='/' then**

**dbms\_output.put\_line(Div(4,4));**

**else**

**dbms\_output.put\_line('Exception');**

**end if;**

**end;**

**/**

**Output:**





**Task:2** Create a function which can convert kelvin into Celsius

**/create or replace function CelsiustoKelvin(num1 in number)**

**return varchar2 is**

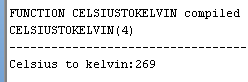
**begin return ('Celsius to kelvin:'||(273 - num1));**

**end;**

**/**

**Select CelsiustoKelvin(4) From dual;**

**Output:**

****

**Lab4**

**ORDBMS –Collection**

**Task 1:** Create an employee record using varray.

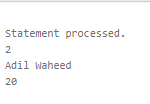
**Code And Output:**

**Create or Replace Type emp\_varray as Varray(3) of varchar(20);**

**declare**

**emp emp\_varray := emp\_varray('2','Adil Waheed','20');**

**begin**

**FOR e IN 1..emp.count LOOP**

**DBMS\_OUTPUT.PUT\_LINE(emp(e));**

**END LOOP;**

**end;**

**/**

**Task 2:** Show the list of week days using varray

**Code And Output:**

**create or replace type week as varray(7) of varchar(10);**

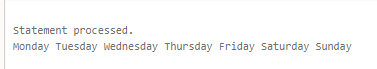
**-- set serveroutput on**

**Declare dayz week:=week('Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday');**

**begin**

**for i in 1..dayz.count**

**loop**

**dbms\_output.put(dayz(i)||' ');**

**end loop;**

**dbms\_output.new\_line;**

**end;/**

**Task 3:** Display students mark sheet with total subject marks using varray.

**Code And Output:**

**Create or Replace Type subs\_ty as object(**

**sub\_name varchar(50),**

**sub\_mark number(3)**

**);**

**Create or Replace Type sub\_v as Varray(3) of subs\_ty;**

**Create Table std(**

**rollno number(10),**

**name varchar(20),**

**marks sub\_v**

**);**

**Insert into std values (12345 ,'Adil Waheed', sub\_v(subs\_ty('English' , 70) , subs\_ty('Urdu' , 80) , subs\_ty('Math' , 100)));**

**set SERVEROUTPUT ON;**

**declare**

**cursor cur is select \* from std;**

**begin**

**FOR s IN cur LOOP**

**DBMS\_OUTPUT.PUT\_LINE('roll no.:' || s.rollno);**

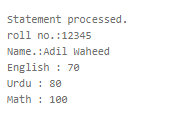
**DBMS\_OUTPUT.PUT\_LINE('Name.:' || s.name);**

**for i in 1..s.marks.count**

**loop**

**DBMS\_OUTPUT.PUT\_LINE(s.marks(i).sub\_name || ' : ' || s.marks(i).sub\_mark);**

**END LOOP;**

**END LOOP;**

**end**;

/

**LAB5**

**ORDBMS –Date Time and package**

**TASK1:**Create a table bahria\_student (Enrollmentno , studentName ,EnrollmentDate ), for all students enrolled for fewer than 200 months, display the enrollment number, EnrollmentDate, number of months enrolled , six-month review date, fırst Friday after enrollment, and last day of the month when enrolled.

**Code & OutPut:**

create table bahria\_student (

Enrollmentno number,

studentName varchar2(4000),

EnrollmentDate date

);

/

INSERT INTO bahria\_student VALUES(1,'Adil',SYSDATE);

INSERT INTO bahria\_student VALUES(2,'Farook','15-MAR-2020');

INSERT INTO bahria\_student VALUES(3,'Hassan','25-May-2020');

INSERT INTO bahria\_student VALUES(4,'kiran','3-AUG-2022');

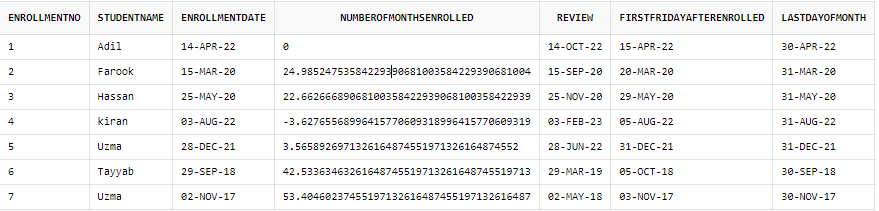
INSERT INTO bahria\_student VALUES(5,'Uzma','28-DEC-2021');

INSERT INTO bahria\_student VALUES(6,'Tayyab','29-SEP-2018');

INSERT INTO bahria\_student VALUES(7,'Uzma','2-Nov-2017');

select Enrollmentno,studentName,EnrollmentDate,MONTHS\_BETWEEN(SYSDATE, EnrollmentDate) numberofmonthsenrolled,

ADD\_MONTHS(EnrollmentDate, 6) REVIEW,NEXT\_DAY(EnrollmentDate,'FRIDAY') FırstFridayAfterEnrolled,LAST\_DAY(EnrollmentDate) LastDayOfMonth

 from bahria\_student;

**TASK2:** Consider the package below named emp\_actions. The package spec declares the following subprograms:

1. Functions hire\_employee
2. Procedure to fire employee

**Code & OutPut:**

create table Employee (

Emp\_no number,

Emp\_name varchar2(4000),

salary number,

HireDate date

)

CREATE OR REPLACE PACKAGE emp\_actions

AS

FUNCTION hire\_employee(Emp\_no number,

Emp\_name varchar2,

salary number,

HireDate date)

RETURN NUMBER;

PROCEDURE fire\_employee(Emp\_id number);

END emp\_actions;

/

CREATE OR REPLACE PACKAGE BODY emp\_actions AS

FUNCTION hire\_employee (Emp\_no number,Emp\_name varchar2,salary number,HireDate date) RETURN NUMBER

IS

new\_emp\_id NUMBER;

BEGIN

INSERT INTO Employee(Emp\_no,Emp\_name,Salary,HireDate)

VALUES (hire\_employee.Emp\_no,hire\_employee.Emp\_name,hire\_employee.salary,hire\_employee.HireDate);

RETURN new\_emp\_id;

END hire\_employee;

PROCEDURE fire\_employee (emp\_id NUMBER) IS

BEGIN

DELETE FROM Employee WHERE Emp\_no = emp\_id;

END fire\_employee;

END emp\_actions;

/

DECLARE

emp\_id NUMBER;

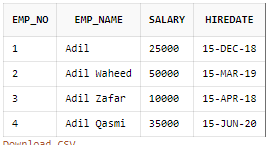
BEGIN

emp\_id:= emp\_actions.hire\_employee(1, 'Adil', 25000 , '15-Dec-2018');

emp\_id:= emp\_actions.hire\_employee(2, 'Adil Waheed', 50000, '15-MAR-2019');

emp\_id:= emp\_actions.hire\_employee(3, 'Adil Zafar', 10000, '15-Apr-2018');

emp\_id:= emp\_actions.hire\_employee(4, 'Adil Qasmi', 35000, '15-Jun-2020');

END;

/

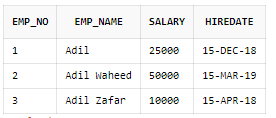
select \* from Employee

**DECLARE**

**emp\_id NUMBER;**

**BEGIN**

**emp\_actions.fire\_employee (4);**

**END;**

**/**

**select \* from Employee**

**LAB6**

**ORDBMS –OBJECT TABLES**

**TASK1:** Create an object table of employee and its department show the Dref and value .

**Code & OutPut:**

create or replace type dept\_ty as object

(

dept\_no number,deptname varchar2(20),dept\_strength number

);

create table depart of dept\_ty;

insert into depart values(30,'IT',20);

insert into depart values(20,'Cs',30);

select \* from depart;

ref(a)

create table refemp

(

eno number ,emp\_name varchar2(20),salary number,dept ref dept\_ty

);

insert into refemp select 45,'shane',45000,ref(a) from depart a where dept\_no=30;

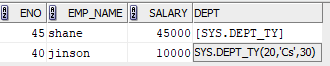
insert into refemp select 40,'jinson',10000,ref(a) from depart a where dept\_no=20;

select \* from depart;

select \* from refemp;

select deref(a.dept) from refemp a where eno=45





**TASK2:** Create an object table for book and authors give the reference and value.

**Code & OutPut:**

select ref(a) from author a where author\_type='Novel Nigar';

create table Book

(

book\_id number ,book\_title varchar2(20),author ref author\_ty

);

insert into Book select 1,'karachi sy Lahore' ,ref(a) from author a where author\_type='Drama Nigar';

insert into Book select 2,'Lahore sy Peshawar' ,ref(a) from author a where author\_type='Drama Nigar';

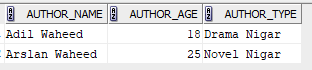
insert into Book select 3,'Peshawar sy karachi' ,ref(a) from author a where author\_type='Novel Nigar';

select \* from author;

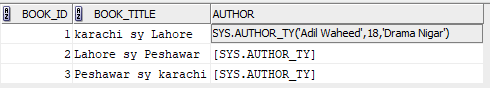
select \* from Book;

--select deref(a.author) from Book a where book\_id=1

ref(a)







**Lab 7**

**Oracle Objects Oriented- Basic components of Oracle objects**

**TASK NO 1:**

* Create an object of Department.
* Create a table of Company which use that department object, while insert some record into the table.
* Display the records.

**OUTPUT:**

create or replace type DepartmentObject as object(

depart\_name varchar2(30),

depart\_id number(5)

);

Create or replace type CompanyObject as object(

Company\_title varchar2(20),

Company\_address varchar2(50),

depart\_details DepartmentObject

);

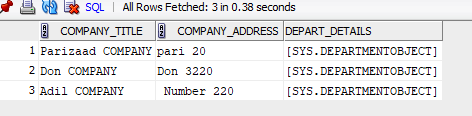
create table CompanyTable of CompanyObject

insert into CompanyTable values('Parizaad COMPANY','pari 20',DepartmentObject('BSE DepartMent',1));

insert into CompanyTable values('Don COMPANY','Don 3220',DepartmentObject('BSCS DepartMent',2));

insert into CompanyTable values('Adil COMPANY',' Number 220',DepartmentObject('Devop DepartMent',3));

select \* from CompanyTable



**TASK NO 2:** Create an object of car(Model, Speed, Color), set its value and implement following member function which:

* 1. Insert Record in table Manufacturer.
  2. Display Record from Manfacturer.

**OUTPUT:**

create or replace type Cars as object(

modell varchar2(20),

speed number,

color varchar2(15),

member function speedIncreasing (inc number) return Cars,

member procedure display,

map member function measure return number);

create or replace type Body Cars AS

member function speedIncreasing (inc number) return Cars is

Begin

return Cars(self.speed+inc,self.modell,self.color);

end speedIncreasing;

member procedure display is

begin

dbms\_output.put\_line('Speed : '||speed );

dbms\_output.put\_line('Color: '||color );

dbms\_output.put\_line('Model : '||modell);

end display;

map member function measure return number is

begin

return (speed);

end measure;

end;

set serveroutput on;

declare

c1 Cars;

c2 Cars;

c3 Cars;

inc number:=10;

Begin

c1 := Cars('Honda',200,'Red');

c2 := Cars('Mehran',80,'Grey');

c3 := Cars('Porche',300,'Orange');

if(c1 > c2) then

c1.display;

else

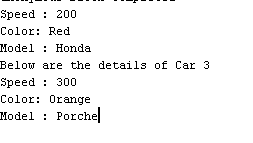
c2.display;

End if;

dbms\_output.put\_line('Below are the details of Car 3');

c3.display;

end;



**Lab 8   
Oracle Objects Oriented - Applying object model to inherited data**

**Task 1:**

Create an object of Mobile which will hold member function display name while name is

set in inherited class of Nokia and Samsung

**OutPut:**

**create or replace type Mobile as object(**

**modell varchar2(10),**

**weight number(5),**

**color varchar2(15),**

**member function CallIn(cal varchar2) return Mobile,**

**member function Messaging(msg varchar2) return Mobile,**

**not final member procedure display**

**)not final;**

**create or replace type Body Mobile AS**

**member function CallIn(cal varchar2) return Mobile is**

**Begin**

**return CallIn(cal);**

**end CallIn;**

**member function Messaging(msg varchar2) return Mobile is**

**Begin**

**return Messaging(msg);**

**end Messaging;**

**member procedure display is**

**begin**

**dbms\_output.put\_line('Weight : '||weight );**

**dbms\_output.put\_line('Color: '||color );**

**dbms\_output.put\_line('Model : '||modell);**

**end display;**

**end;**

**create or replace type samsung under Mobile**

**(size1 varchar2(20),**

**OVERRIDING member procedure display);**

**create or replace type Nokia under Mobile**

**(size1 varchar2(20),**

**OVERRIDING member procedure display);**

**create or replace type body samsung as**

**overriding member procedure display is**

**begin**

**dbms\_output.put\_line('Weight : '||weight );**

**dbms\_output.put\_line('Color: '||color );**

**dbms\_output.put\_line('Model : '||modell);**

**dbms\_output.put\_line('Size : '||size1);**

**end display;**

**end;**

**create or replace type body Nokia as**

**overriding member procedure display is**

**begin**

**dbms\_output.put\_line('Weight : '|| weight );**

**dbms\_output.put\_line('Color: '|| color );**

**dbms\_output.put\_line('Model : '|| modell);**

**dbms\_output.put\_line('Size : '|| size1);**

**end display;**

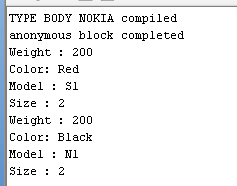
**end;**

**set serveroutput on;**

**declare**

**c1 samsung;**

**c2 Nokia;**

 **inc number:=10;**

**Begin**

**c1 := samsung('S1',200,'Red','2');**

**c2 := Nokia('N1',200,'Black','2');**

**c1.display();**

**c2.display();**

**end;**