

## Lab 1

# INTRODUCTION TO DENORMALIZATION

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

order		User	
Order id	No (pk)	User id	No (pk)
Order name	Varchar2	User name	Varchar2
		Order id	No (fk)

**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

rollno	name	branch	hod	office_tel
401	Akon	CSE	Mr. X	53337
402	Bkon	CSE	Mr. X	53337
403	Ckon	CSE	Mr. X	53337
404	Dkon	CSE	Mr. X	53337

- **Solution: [DENORMALIZED]:**

<pre>select * from Denormalized_table select * from Emp_table, Std where Std_Empl_id_Pk=Emp_Id SELECT * FROM Emp_table INNER JOIN Std</pre>							
<b>Results</b>							
	Std_key	Std_Name	Std_Rollno	Std_Empl_id_Pk	Emp_Name	branch_Name	branch_key
1	1	Adil Waheed	12	1	Cheema	CSS	1
2	2	Cheema Don	15	1	Maa'im Ramshaa	BSE	2

- **NORMALIZED**

SQLQuery3.sql - (lo...lization (sa (58)))

```

select * from Std
select * from Branch_table
select * from Emp_table
select * from Emp_table,Std where Std_Empl_id_Pk=Emp_Id
SELECT *
FROM Emp_table
INNER JOIN Std
ON Emp_table.Emp_Id = Std.Std_Empl_id_Pk;
  
```

Results

Std_key	Std_Name	Std_Rollno	Std_Empl_id_Pk
1	adil	12	1
2	waheed	12	1

branch_key	branch_Name	DateTime
1	CSE	Mar 11 2022 3:51PM
2	BSE	Mar 11 2022 3:52PM
3	CSS	Mar 11 2022 3:53PM

Emp_Id	Emp_Name	Emp_phone
1	ali	1234
2	Ahmed	1234

Emp_Id	Emp_Name	Emp_phone	Std_key	Std_Name	Std_Rollno	Std_Empl_id_Pk
1	ali	1234	1	adil	12	1
2	1	ali	2	waheed	12	1

Emp_Id	Emp_Name	Emp_phone	Std_key	Std_Name	Std_Rollno	Std_Empl_id_Pk
1	1	ali	1	adil	12	1
2	1	ali	2	waheed	12	1

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

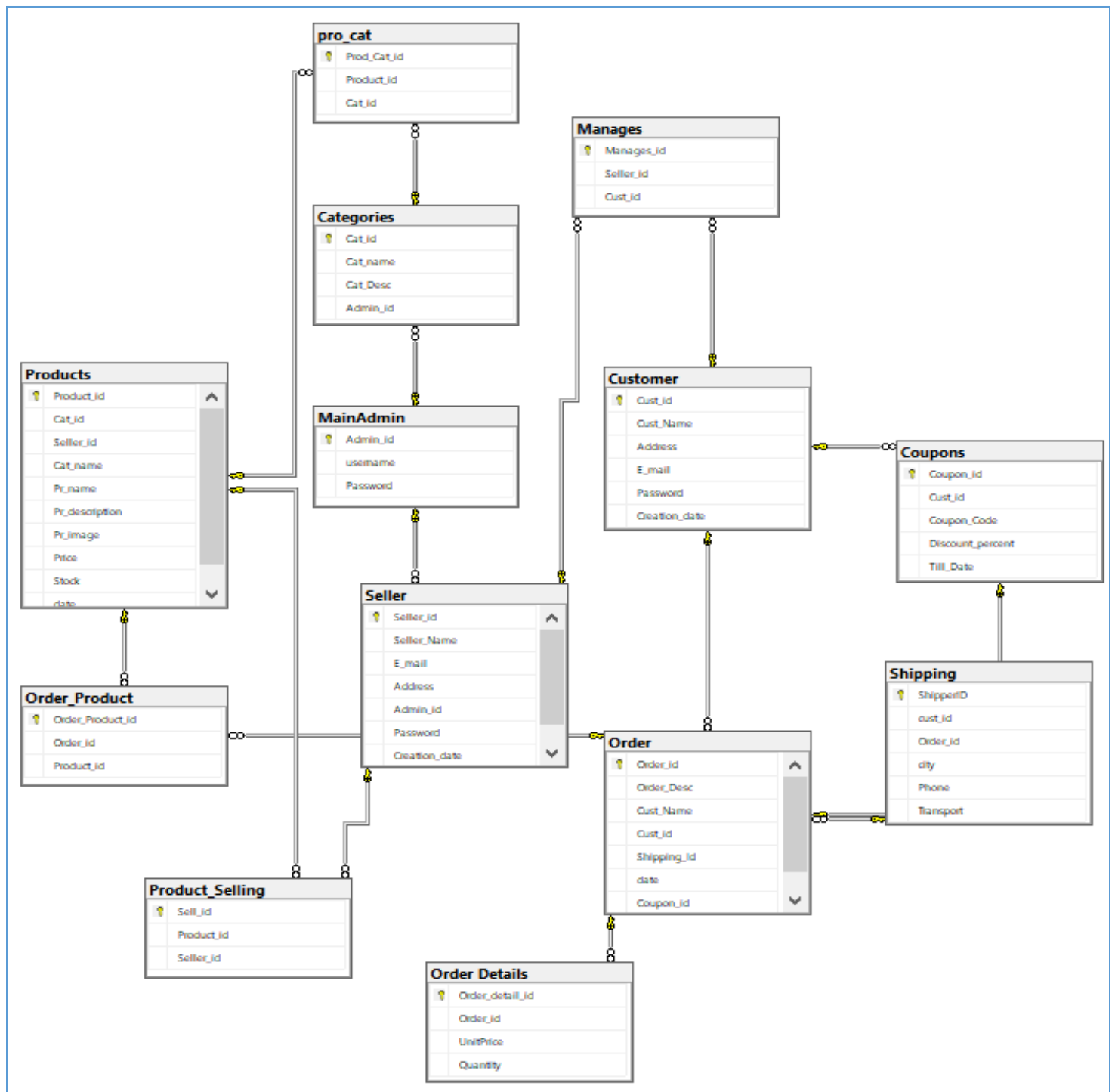
1. Check the database and tables architecture
2. Normalize if any table is not normalize
3. Check for table who need to be denormalized.

**Solution:**

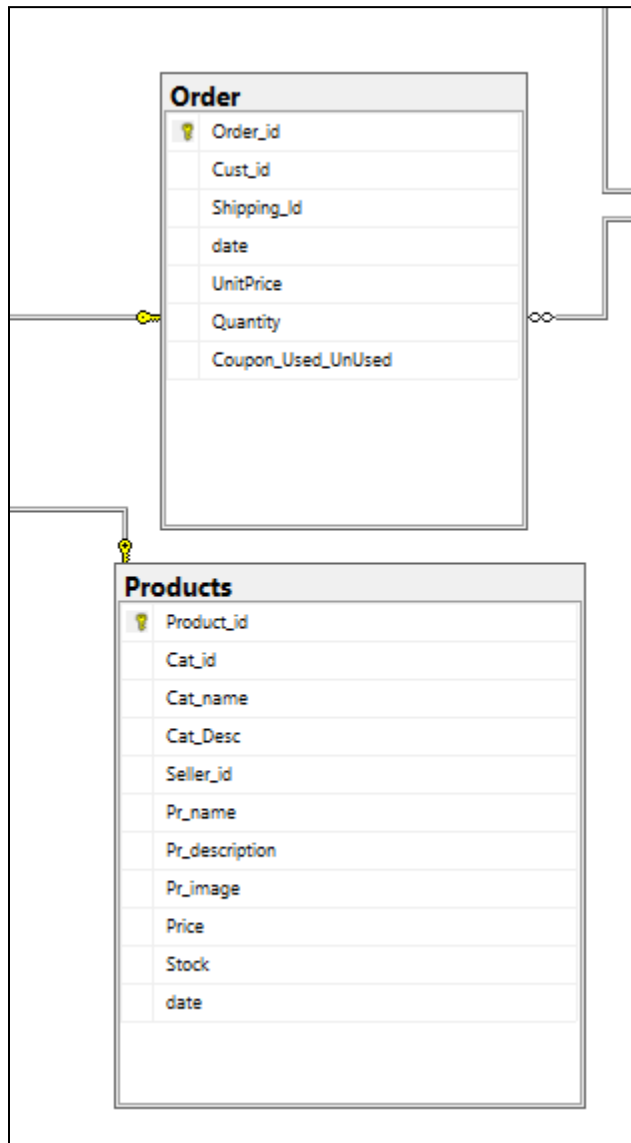
**This Database Already Normalized**

**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
);

```

TEACHER_NAME	SUBJECT	SEMESTER	TEACGER_ID
Maam Ramshaa Masood	Advance Database	6A	2
Sir Majid	SQE	6A	3
Sir Faisal	Advance Database	6A	4
Maam Ramshaa Masood	Data Mining	6A	1

NAME	ADDRESS	PHONE_NO	FATHER_NAME	STD_ID
Arslan waheed	Majeed Sre	3212347886	Abdul waheed	2
Adil waheed	Majeed Sre	3212347886	Abdul waheed	1
Muhammad Tayyab	Majeed Sre	3212347886	Abdul waheed	3

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

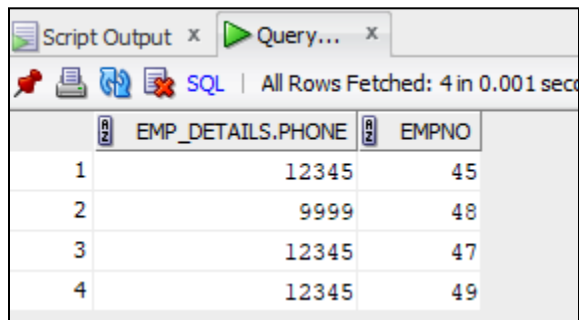
TEACHER_NAME	SUBJECT	SEMESTER	TEACGER_ID
Sir Majid	SQE	6A	3
Sir Faisal	Advance Database	6A	4
Maam Ramshaa Masood	Data Mining	6A	1

**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:



	EMP_DETAILS.PHONE	EMPNO
1	12345	45
2	9999	48
3	12345	47
4	12345	49

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

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00

Code:

ID	NAME	AGE	ADDRESS	SALARY
4	Chaitali	25	Mumbai	6500
3	kiran	23	Kota	2000
1	Ramesh	32	Ahmedabad	2000
2	Khilan	25	Delhi	1500
5	Hardik	25	Bhopal	8500
6	Komal	22	Mp	4500

[Download CSV](#)

```
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'
```

ID	NAME	AGE	ADDRESS	SALARY
4	Chaitali	25	Mumbai	6500
3	kiran	23	Kota	2000
1	Ramesh	32	Ahmedabad	2000
2	Mehvish	25	Delhi	1500
5	Hardik	25	Bhopal	8500
6	Komal	22	Mp	4500

```

        6, 'Komal', 22, 'Mp', '4500'
    );
/
select * from Customer;

UPDATE Customer SET name = 'Mehvish' WHERE id=2;
DELETE FROM Customer t WHERE t.id = '5';

```

ID	NAME	AGE	ADDRESS	SALARY
4	Chaitali	25	Mumbai	6500
3	kiran	23	Kota	2000
1	Ramesh	32	Ahmedabad	2000
2	Mehvish	25	Delhi	1500
6	Komal	22	Mp	4500

**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;/

```

```

anonymous block completed
Addition Of two Number Is=100
Subtraction Of two Number Is=10
Multiplication Of two Number Is=2475
Division Of two Number Is=1.22222222222222

```

## 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:**

anonymous block completed Division:1	anonymous block completed Multiplication:16 	anonymous block completed Addition:8
anonymous block completed Subtraction:0		

**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:**

FUNCTION CELSIUSTOKELVIN compiled CELSIUSTOKELVIN(4) ----- Celsius to kelvin:269
---

## 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;
/

```

```

Statement processed.
2
Adil Waheed
20

```

**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;/

```

```

Statement processed.
Monday Tuesday Wednesday Thursday Friday Saturday Sunday

```

**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;

/

```

```

Statement processed.
roll no.:12345
Name.:Adil Waheed
English : 70
Urdu : 80
Math : 100

```

## 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, first 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')  
FirstFridayAfterEnrolled, LAST_DAY(EnrollmentDate) LastDayOfMonth  
  
from bahria_student;
```

ENROLLMENTNO	STUDENTNAME	ENROLLMENTDATE	NUMBEROFMONTHSENROLLED	REVIEW	FIRSTFRIDAYAFTERENROLLED	LASTDAYOFMONTH
1	Adil	14-APR-22	0	14-OCT-22	15-APR-22	30-APR-22
2	Farook	15-MAR-20	24.98524753584229390681003584229390681004	15-SEP-20	20-MAR-20	31-MAR-20
3	Hassan	25-MAY-20	22.66266689068100358422939068100358422939	25-NOV-20	29-MAY-20	31-MAY-20
4	kiran	03-AUG-22	-3.62765568996415770609318996415770609319	03-FEB-23	05-AUG-22	31-AUG-22
5	Uzma	28-DEC-21	3.5658926971326164874551971326164874552	28-JUN-22	31-DEC-21	31-DEC-21
6	Tayyab	29-SEP-18	42.53363463261648745519713261648745519713	29-MAR-19	05-OCT-18	30-SEP-18
7	Uzma	02-NOV-17	53.40460237455197132616487455197132616487	02-MAY-18	03-NOV-17	30-NOV-17

**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.HireD
ate);
        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

EMP_NO	EMP_NAME	SALARY	HIREDATE
1	Adil	25000	15-DEC-18
2	Adil Waheed	50000	15-MAR-19
3	Adil Zafar	10000	15-APR-18
4	Adil Qasmi	35000	15-JUN-20

[Download CSV](#)

DECLARE

emp\_id NUMBER;

BEGIN

emp\_actions.fire\_employee (4);

END;

/

select \* from Employee

EMP_NO	EMP_NAME	SALARY	HIREDATE
1	Adil	25000	15-DEC-18
2	Adil Waheed	50000	15-MAR-19
3	Adil Zafar	10000	15-APR-18

## **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

```

DEPT_NO	DEPTNAME	DEPT_STRENGTH
30	IT	20
20	Cs	30

DEREF(A.DEPT)	
1	SYS.DEPT_TY(30,'IT',20)

ENO	EMP_NAME	SALARY	DEPT
45	shane	45000	[SYS.DEPT_TY]
40	jinson	10000	SYS.DEPT_TY(20,'Cs',30)

**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)

```

AUTHOR_NAME	AUTHOR_AGE	AUTHOR_TYPE
Adil Waheed	18	Drama Nigar
Arslan Waheed	25	Novel Nigar

REF(A)
SYS.AUTHOR_TY('Arslan Waheed',25,'Novel Nigar')

BOOK_ID	BOOK_TITLE	AUTHOR
1	karachi sy Lahore	SYS.AUTHOR_TY('Adil Waheed',18,'Drama Nigar')
2	Lahore sy Peshawar	[SYS.AUTHOR_TY]
3	Peshawar sy karachi	[SYS.AUTHOR_TY]

## 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

```

	COMPANY_TITLE	COMPANY_ADDRESS	DEPART_DETAILS
1	Parizaad COMPANY	pari 20	[SYS.DEPARTMENTOBJECT]
2	Don COMPANY	Don 3220	[SYS.DEPARTMENTOBJECT]
3	Adil COMPANY	Number 220	[SYS.DEPARTMENTOBJECT]

**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 Manufacturer.

**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;

```

```

Speed : 200
Color: Red
Model : Honda
Below are the details of Car 3
Speed : 300
Color: Orange
Model : Porche

```

## 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 : '||model1);
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 : '||model1);

```

```

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;

```

```

TYPE BODY NOKIA compiled
anonymous block completed
Weight : 200
Color: Red
Model : S1
Size : 2
Weight : 200
Color: Black
Model : N1
Size : 2

```