-----Q1

CREATE OR REPLACE TYPE Empl\_typ AS OBJECT

(

empNo INT,

empName VARCHAR2(50),

salary FLOAT,

designation VARCHAR2(50)

);

/

CREATE OR REPLACE TYPE Member\_typ AS OBJECT

(

team\_member REF Empl\_typ

);

/

CREATE TYPE Member\_nt AS TABLE OF Member\_typ;

/

CREATE TYPE Project\_typ AS OBJECT

(

projNo INT,

pname VARCHAR2(50),

members Member\_nt,

mgr REF Empl\_typ

);

/

CREATE TABLE Employees OF Empl\_typ(PRIMARY KEY(empNo));

CREATE TABLE Projects OF Project\_typ(PRIMARY KEY (projNo), mgr REFERENCES Employees)

NESTED TABLE members STORE AS member\_tab;

/

INSERT INTO Employees VALUES (1, 'Sarath', 10000.0, 'Technician');

INSERT INTO Employees VALUES (2, 'Sampath', 12000.0, 'Sales Officer');

INSERT INTO Employees VALUES (3, 'Viraj', 18000.0, 'Software Developer');

INSERT INTO Employees VALUES (4, 'Tusith', 22000.0, 'Director');

INSERT INTO Employees VALUES (5, 'Nimali', 15000.0, 'Technician');

INSERT INTO Projects VALUES (10, 'Cabling Project',

member\_nt(

Member\_typ((select ref(e) from Employees e where e.empNo=1)),

Member\_typ((select ref(e) from Employees e where e.empNo=5))),

null);

/

INSERT INTO Projects VALUES (12, 'Data Warehousing Proj',

member\_nt(

Member\_typ((select ref(e) from Employees e where e.empNo=2)),

Member\_typ((select ref(e) from Employees e where e.empNo=3)),

Member\_typ((select ref(e) from Employees e where e.empNo=4))),

(select ref(e) from Employees e where e.empNo=2)

);

/

**--Print the project number, project name and project manager’s salary. The results should be printed in descending order of salary values.**

select p.projNo, p.pname, p.mgr.salary

from Projects p

order by p.mgr.salary DESC

/

**--Write a member method (called Budget) that finds returns the total salary of all members in the project.**

alter type Project\_typ

Add member function Budget

return float

cascade

/

create or replace type body Project\_typ as member function Budget return float is

pBudget float;

begin

select sum(m.team\_member.salary) into pBudget

from table(self.members) m;

return pBudget;

end;

end;

/

--call Function

select p.Budget()

from Projects p

**--Use the method created above to find the project with the least budget greater than 20000. Print the** projNo.

SELECT p.projNo

FROM Projects p

WHERE p.Budget() > 20000 and ROWNUM <= 1

ORDER BY p.Budget();

/

--------Q2

create type account\_type\_typ as object (

code char(4),

name varchar2(50),

interest float,

min\_bal float

);

/

create type account\_typ as object (

account\_no char(10),

account\_ty ref account\_type\_typ,

balance float

);

/

create type account\_tlb as table of account\_typ;

/

create table account\_types of account\_type\_typ (code primary key);

create type customer\_typ as object (

id char(10),

name varchar2(50),

accounts account\_tlb

);

/

create table customers of customer\_typ(id primary key)

nested table accounts store as ntlb\_accounts;

/

insert into account\_types values ('SAVS', 'General Savings', 5.24, 100.0);

insert into account\_types values ('CHEQ', 'General Checking', 0.0, 0);

insert into account\_types values ('HAPS', 'Hapan Savings', 6.26, 20000.0);

insert into customers values (

'781250401V',

'Sampath Weerasinghe',

account\_tlb(

account\_typ('1122300004',

(select ref(s) from account\_types s where s.code = 'SAVS'), 5700.00),

account\_typ('2334453124',

(select ref(s) from account\_types s where s.code = 'CHEQ'), 2300.00)

)

);

/

insert into customers values (

customer\_typ (

'99334453V', 'Dulani Pieris',

account\_tlb(

account\_typ('3445663321',

(select ref(s) from account\_types s where s.code = 'HAPS'), 19045.06),

account\_typ('5000022123',

(select ref(s) from account\_types s where s.code = 'SAVS'), 55235.00)

)

)

);

/

**--Deposit Rs. 1200/- to the account (Acc # 1122300004) of Mr. Sampath Weerasignhe (ID # 781250401V)**

update table( select c.accounts from customers c where c.id='781250401V')a

set a.balance = a.balance + 1200.00

where a.account\_no = '1122300004'

/

**--Write a member function (called TotBal) that returns the total balance of a customer’s accounts.**

alter type customer\_typ

Add member function TotBal

return float

cascade

/

create or replace type body customer\_typ as member function TotBal return float is

totBalance float;

begin

select sum(ac.balance) into totBalance

from table(self.accounts) ac;

return totBalance;

end TotBal;

end;

/

--call Function

select c.TotBal()

from customers c

**--Write a query to print the names of customers and their total balances (using the function created in question 2)**

select c.name AS CustomerName, c.TotBal() AS TotalBalance

from customers c