Sailors Database

Sailors (sid, sname, age, rating)

boats (bid, bname, color)

reserves (sid, bid, day)

Creation of tables:

```
SQL> create table sailors (sid number(2) primary key, sname varchar2(20)
     not null, age number(2) not null, rating number(2) not null);
```

Output:

```
SQL> create table boats (bid number(3) primary key, bname varchar2(20) not null,
                         color varchar2(15));
```

Output:

```
SQL>create table reserves (sid number(2) references sailors, bid number(3)
          references sailors boats, day date not null);
```

Output:

Compute the following queries : (1)

a) Find the sailors who reserved two boats -

```
SQL> select sname from sailors where sid = (select sid from reserves group by
     sid having count(sid) = 2);
```

Output:

SNAME
Horatio

b) Include check constraint on rating such that rating > 150

```
SQL> alter table sailors add constraint c_k check (rating > 150);
```

c) Find the sailors who reserved red and green boats.

```
SQL> select distinct s.sname from sailors s, reserves r1, boats b1, reserves r2,
     boats b2 where s.sid = r1.sid and r1.bid = b1.bid and s.sid = r2.sid and
     r2.bid = b2.bid and b1.color = 'Red' and b2.color = 'Green';
```

Output:

SNAME
Brutus
Dustin

d) Find the boats which were not reserved by any sailors.

```
SQL> select bname from boats where bid in (select bid from boats) and
     bid not in (select bid from reserves);
```

Output: No rows selected

e) find the name of the oldest sailor.

```
SQL> select sname from sailors where age = (select max(age) from sailors);
```

Output:

```
SNAME
Bob
```

Write & Compute PL/SQL:

a) create a package with one procedure & one function.
Procedure displays the name of boats for the given boat id.
function (displays) returns the color of boat for the given boat id.

Package specification:

```
create or replace package pack as
procedure displayname(id in boats.bid%type);
function displaycolor(id in boats.bid%type) return varchar2(15);
end;
/
```

O/P: package created

Package body:

```
create or replace package body pack is
procedure displayname(id in boats.bid%type) as
name boats.bname%type;
begin
    select bname into name from boats where bid=id;
    dbms_output.put_line('Boat name:'||name);
end displayname;
function displaycolor(id in boats.bid%type)
return varchar2(15) as
x varchar2(15);
//name boats.bname%type; col boats.color%type;
begin
    select color into col from boats where bid = id;
    x:= col;
    return(x);
end displaycolor;
end pack;
/
```

O/P: package body created

boats) and
m sailors);

in program:

```
declare
        id  boats.bid%type := '&id';
        res boats.color%type
begin
        pack.displayname(id);
        res := pack.displaycolor(id);
        dbms_output.put_line('Boat color:'||res);
end;
```

Output:

Enter value for id: 103
Boat name: clipper
Boat color: Green

PL/SQL procedure successfully completed.

Compute the following Queries: (11)

a) Find the color of boat which is present for 1 boat

```
SQL>select color from boats group by color having count(color)=1;
```

Output:

COLOR
Green
Blue

b) Find the sailor whose age is greater than average age of sailors.

```
SQL>select sname from sailors where age>(select avg(age) from sailors);
```

Output:

SNAME
Dustin
Brutus
Horatio
Bob

c) Find the sailors who reserved a red boat.

```
SQL> select distinct sname from sailors s, reserves r, boats b where s.sid=r.sid
     and r.bid = b.bid and b.color = 'Red';
```

Output:

SNAME
Dustin
Brutus
Horatio

d) Find the sailors whose name contains five charecters

```
SQL>select sname from sailors where length(sname)=5;
```

Output:

SNAME
Rusty
Zorba

e) Find the name of sailor whose rating is maximum.

```
SQL> select sname from sailors where rating = (select max(rating) from
```

Output:

```
SNAME
Rusty
Zorba
```

Write & Compile PL/SQL:

a) Find the name of boats of given color

```
declare
        col boats.color%type := '&col';
        name boats.bname%type;
begin
        select bname into name from boats where color=col;
        dbms_output.put_line ('Boat name :'||name);
end;
```

Output:

```
Enter value for color: Blue
Boat name : Interlake
```

PL/SQL procedure successfully completed.

b) Create a procedure to find the no. of boats reserved by a given sailor.

Program:

```
create or replace procedure numboats (name in sailors.sname%type) as
num number(3);
begin
      select count(sid) from reserves r group by sid having
      sid = (select sid from sailors where sname='Dustin')
      into num;
      dbms_output.put_line ('Number of boats reserved by'||snam
                            'are :'||num);
end;
```

Output:

```
Enter value for
> exec numboats('Dustin');
Number of boats reserved by Dustin are : 4
PL/SQL procedure successfully completed.
```

Compute the following Queries: (3)

a) Find the sailors whose rating is greater than 3.

SQL> select sname from sailors where rating > 3;

Output:

SNAME
Dustin
Brutus
Rusty

b) update the rating of sailors as 4 who belong to guntur.

SQL> update sailors set rating = 4 where loc = 'Guntur'.

c) Find the sailors who reserved three boats.

SQL> select sid, sname from sailors where sid = (select sid from reserves group by sid having count(sid) = 3);

Output:

SID	SNAME
31	Brutus

d) find the boats which were reserved by 2 sailors.

SQL> select bid, bname from boats where bid in (select bid from reserves group by bid having count(bid) = 2);

Output:

BID	BNAME
101	Interlake
104	marine

e) find the name of sailors whose age is greater than aveggerage age of all sailors.

Output:

SNAME
Dustin
Brutus
Horatio
Bob

Write & Compute PL/SQL:

a) Create a package with one procedure and one function.
Procedure displays the color of boat for the given bid.
Function returns the name of boat for the given boat id.

Package Specification:

```
Create or replace package pack as
procedure displaycolor (id in boats.bid%type);
function displayname (id in boats.bid%type)
return varchar2(15);
end;
```

O/p: Package created.

Package Body:

```
create or replace package body pack is
Procedure displaycolor (id in boats.bid%type) as
Color  boats.color%type;
begin
      select color into color from boats where bid=id;
      dbms_output.put_line ('Boat color:'||colo);
end displaycolor;
function displayname (id in boats.bid%type)
return varchar2(15) as
 name boats.bname%type;
  X  varchar2(15);
begin
      select bname into name from boats where bid=id;
      X:=name;
      return(X);
end displayname;
end pack;
```

O/p: package body created.

Main program:

```
declare
id  boats.bid%type:='&id';
res  boats.bname%type;
begin
      pack.displayColor(id);
      res:=pack.displayname(id);
      dbms_output.put_line ('Boat name:'||res);
end;
```

Output:

Enter value for id: 102
Boat color: Interlake
Boat name: Red.

mpute the following Queries: (10)

a) Find the color of boat which is present for 2 boats.

SQL> select color from boats group by color having count(color)=2;

Output:
COLOR
Red

b) Find the sailor whose age is less than average age of sailors

SQL> select sname from sailors where age < (select avg(age) from sailors);

Output:
SNAME
Rusty
Zorba
Art
Brutus

c) Find the sailors who reserved two boats

SQL> select sname from sailors where sid = (select sid from reserves group by sid having count(sid) = 2);

Output:
SNAME
Horatio

d) Find the sailors whose name contains 3 charecters

SQL > select sname from sailors where length(sname) = 3;

Output:
SNAME
Art
Bob

e) find the name of sailors whose rating is minimum.

SQL > select sname from sailors where rating = (select min(rating) from sailors);

Output:
SNAME
Brutus

Write & Compute PL/SQL:

a) Find the bids of boats for a given color.

```
declare
        col boats.color%type = '&col';
        id boats.bid %type;
begin
        select bid into id from boats where color = col ;
        dbms_output.put_line('Boat id for given color'||col||'is : '|| id)
end;
```

Output:
Enter value for col : Green
Boat id for given color Green is 103.

PL/SQL procedure successfully completed.

b) create a procedure to find the no. of boats reserved by a given sail

Program:

```
create or replace procedure numboats (name in boats.bname%type)
as
res number(2);
begin
    Select count(sid) from reserves group by sid having sid = (select
    from sailors where sname = 'Dustin') into res;
    dbms_output.put_line ('No. of boats reserved by' ||name|| 'a
                          ||res);
end;
```

Output:

~~Enter value~~

```
>exec numboats ('Dustin');
Nb of boats reserved by Dustin are 4.
PL/SQL procedure successfully completed.
```

Employee Database

Employee (empid, ename, city, mgrid, salary, deptid, comm)

Department (deptid, dname, location)

(Hint: (mgrid) ⊆ (empid))

Creation of tables:

```
SQL> create table emp (empid number(4) primary key, ename varchar2(15)
     not null, city varchar2(15) not null, mgrid number(4), salary number(
     not null, deptid number(2) not null, comm number(5,3));

QL> create table dept (deptid number(2) primary key, dname varchar2(1
    not null, loc varchar2(15) not null);
```

pute the following queries: (2)

Add new column to the table employee.

SQL> alter table emp add (dob date);

output: Table altered.

b) Find the department with no employees.

SQL> select dname from dept where deptno in (select deptno from dept) and deptno not in (select deptno from emp);

output: DNAME
OPERATIONS

c) Find the department with one employee.

SQL> select dname from dept where deptno in (select deptno from emp group by deptno having count(deptno)=1);

output: no rows selected.

d) Find the department located at Hyderabad.

SQL> select dname from dept where loc='Hyderabad';

output: RESEARCH

e) Find details of managers.

SQL> select w.mgr as mgrid, m.ename as mgrname from emp w, emp m where w.mgr = m.empno;

Output:

MGRID	MGRNAME
7566	JIM
7698	BLAKE
7782	CLARK
7788	SCOTT

Write & Compute PL/SQL:

a) Find the names of employees whose sal is > 3000.

Program:

```
declare
        cursor s is select ename from emp where sal>3000;
        x s%rowtype;
begin
        open s;
        loop
            fetch s into x;
            exit when s%notfound;
            dbms_output.put_line('Employee name :'||x.ename);
        end loop;
        close s;
end;
```

Output: /

Employee Name: JIM
Employee Name: BLAKE
Employee Name: SCOTT
Employee NAME: FORD

b) create a function to return the salary of an employee for the given

Program: (Creating function)

```
create or replace function displaysal(id in emp.empid%type)
return varchar2(15) as
sal emp.salary%type;
x varchar2(15);
begin
    select salary into sal from emp where empid=id;
    x:=sal;
    return(x);
end;
```

Output:

Enter

Calling function:

```
declare
id emp.empid%type:='&id';
res varchar2(15);
begin
    res:=displaysal(id);
    dbms_output.put_line('Salary is:'||res);
end;
```

Output:

1) Enter value for id: 7369
Salary is 800

2) Enter value for id: 7934
Salary is 7934

PL/SQL procedure successfully completed.

pute the following Queries: (5)

a) Find the employees whose salary is greater than his departments average salary.

SQL > select ename from emp where sal > (select avg(sal) from emp group by deptno);

output: ENAME
SMITH

b) Include check constraint over the column comm such that comm is > 50

SQL > alter table emp add constraint c_k check (comm > 50);

c) Find the employees who are not managers

SQL > select ename from emp where empno not in (select mgr from emp where mgr is not null);

output: ENAME
SMITH
ALLEN
WARD
MARTIN
TURNER

d) Find the employee who work in place where he lives.

SQL > select ename from emp where city = (select loc from dept);

output: ENAME
TURNER

e) Find the name of employees whose comm is null.

SQL > select ename from emp where comm is null;

Output: ENAME
SMITH
JIM
BLAKE
CLARK
SCOTT
JACK
ADAMS
JAMES
FORD
MILLER

Write & Compute PL/SQL:

a) create a package with one function & one procedure.
function returns the location of department for the given department
Procedure displays the no. of employees in the department for given department id.

Package Specification:

```
Create or replace package pack as
function displaylocation(id in dept.deptno%type) return varchar2(15);
procedure numemp(id in dept.deptno%type);
end;
```

O/P: Package created

Package body:

```
create or replace package body pack is
function displaylocation(id in dept.deptno%type) return varchar2(15) as
res varchar2(15);
loct dept.loc%type;
begin
    select loc into loct from dept where deptno=id;
    res:=loct;
    return(res);
end displaylocation;
procedure numemp(id in dept.deptno%type) as
result number(2);
begin
    select count(deptno) from emp group by deptno having deptno=id into result;
    dbms_output.put_line('No. of employees are :'||result);
end numemp;
end pack;
/
```

O/P: package body created.

Main program:

```
declare
id dept.deptno%type;='&id';
res varchar2(15);
begin
   res:=pack.displaylocation(id);
   dbms_output.put_line('Location of dept'||id||'is'||res);
    pack.numemp(id);
end;
```

output:

Enter value for id : 30
Location of dept 30 is chicago
No. of employees are 7
PL/SQL procedure successfully completed.

pute the following Queries: (8)

a) Update the city of employee as guntur whose department is 10.

SQL> update emp set city = 'Guntur' where deptno = 10.

output: 1 row updated.

b) Find the department with three employees

SQL> select dname from dept where deptno in (select deptno from emp group by deptno having count(deptno) = 3);

output: DNAME
RESEARCH

c) Find the department with one employee.

SQL> select dname from dept where deptno in (select deptno from emp group by deptno having count(deptno) = 1);

Output: no rows selected.

d) Find the department located at Hyderabad & Guntur

SQL> select dname from dept where loc = 'Hyderabad' and loc = 'Guntur';

e) Find the department with no employees.

SQL> select dname from dept where deptno in (select deptno from dept) and deptno not in (select deptno from emp);

output: DNAME
OPERATIONS

Write & Compute PL/SQL:

a) find the names of employees whose salary is < 1000

Program:

```
declare
    cursor s is select ename from emp where sal<1000;
    x s%rowtype;
    begin
        open s;
        loop
            fetch s into x;
            exit when s%.notfound;
            dbms_output.put_line('Employee name:'||x.ename);
        end loop;
        close s;
    end;
```

output: Employee name: SMITH

PL/SQL procedure successfully completed.

b) create a function to return the name of an employee for given em

Creating function:

```
create or replace function displayname (id in emp.empid%type)
return varchar2(15)
as
name empename%type;
x varchar2(15);
begin
    select ename into name from emp where empid=id;
    x:=name;
    return(x);
end;
```

Calling function:

```
declare
id emp.empid%type:='&id';
res varchar2(15);
begin
    res:=displayname(id);
    dbms_output.put_line('Name of Employee is:'||res);
end;
```

Output:

```
Enter value for id: 7369
Name of Employee is: SMITH
PL/SQL procedure successfully completed.
```

ompute the following Queries: (11)

) find the employee with highest salary

```
QL> select ename from emp where sal=(select max(sal) from emp);
```

tput:

```
ENAME
JACK
```

Include check constraint over the column salary such that salary>1500

```
L> alter table emp add constraint c_t check (salary>1500);
```

tput: table altered

ind the employee who are managers

```
> select ename from emp where empno in (select mgr from emp where mgr is
```

ut: ENAME

emp

d) find the employees who works in Guntur.

SQL> select ename from emp e, dept d where e.deptno = d.deptno and d.loc = 'Guntur';

output:
ENAME
WARD
SMITH
TURNER

e) find the names of the employees whose commission is not null.

SQL> select ename from emp where comm is not null;

output:
ENAME
ALLEN
WARD
MARTIN
TURNER

Write & Compute PL/SQL:

a) Create a package with one function & one procedure.
function returns the name of department for given deptid
Procedure displays the no. of employees in department for given deptid.

Package Specification:

```
create or replace package pack as
function displayname(id in dept.deptno%type) return varchar2(15);
procedure numemp(id in dept.deptno%type);
end;
/
```

output: package created.

Package Body:

```
create or replace package body pack is
function displayname (id in dept.deptno%type) return varchar2(15) as
name dept.dname%type;
 x varchar2(15);
begin
      select dname into name from dept where id = deptno;
       x := name;
       return(x);
end displayname;
procedure numemp(id in dept.deptno%type) as
result number(2);
begin
      select count(deptno) from emp group by deptno having deptno = &d into result;
      dbms_output.put_line('No. of employees are:'||result);
end numemp;
end pack;
/
```

output: package body created

Main program:

```
declare
        id dept.deptno%type:='&id';
        res varchar2(15);
        begin
            res:=pack.displayname(id);
            dbms_output.put_line('Name of dept:'||res);
            pack.numemp(id);
        end;
```

Output:

```
Enter value for id: 30
Name of dept: SALES
No. of employees are 7
PL/SQL procedure successfully completed
```

Suppliers Database:

Supplier (sid, sname, city, status)
parts (pid, pname, color, weight)
supply (sid, pid, quantity)

Creation of tables:

```
SQL> create table suppliers(sid varchar2(2) primary key, sname varchar2(10)
     not null, city varchar2(10) not null, status varchar2(10) not null);

SQL> create table parts (pid varchar2(2) primary key, pname varchar2(10)
     not null, color varchar2(10) not null, weight varchar2(5) not null);

SQL> create table supply (sid varchar2(2) not null references suppliers,
                          pid varchar2(2) not null references parts,
                          quantity number(3) not null);
```

pute the following Queries: (3)

Find the parts with maximum weight

SQL> select pname from parts where weight = (select max(weight) from parts);

Output:
PNAME
Cog

b) Include the check constraint on quantity such that quantity > 50.

SQL> alter table supply add constraint

Output: Table altered.

c) Find the part names for colors red & green

SQL> select pname from parts where color in ('Red', 'Green');

Output:
PNAME
Nut
Bolt
Screw
Cog

d) find the parts which were supplied by single suppliers.

SQL> select pname from parts where pid in (select pid from supply group by pid having count(pid) = 1);

Output:
PNAME
Screw
Cog

e) Find the parts which are not supplied by any supplier.

SQL> select pname from parts where pid not in (select pid from supply);

Output:
PNAME
Nail

Create & Compute PL/SQL:

a) create a package with one procedure and one function.
procedure finds the city for a given supplier.
function returns the weight of part for the given part id.

Package Specification:

```
Create or replace package pack as
 procedure displaycity (name in suppliers.sname%type);
function displayweight (id in parts.pid%type) return varchar2 (5);
end;
/
```

Output: Package created.

Package Body:

```
create or replace package body pack is
procedure displaycity (name in suppliers.sname%type) as
loc suppliers.city%type;
begin
    select city into loc from suppliers where sname=name;
    dbms_output.put_line('City for given supplier '||name||' is :'||loc);
end displaycity;
function displayweight(id in parts.pid%type) return varchar2(5) as
wt parts.weight%type;
x  varchar2(5);
begin
    select weight into wt from parts where pid=id;
    x:=wt;
    return(x);
end displayweight;
end pack;
/
```

O/p: Package body created.

Main Program:

```
declare
    name suppliers.sname%type:='&name';
    id   parts.pid%type:='&id';
    res  varchar2(3);
begin
    pack.displaycity(name);
    res:=pack.displayweight(id);
    dbms_output.put_line('Weight of given part is :'||res);
end;
```

Output:

Enter value for name: Smith
Enter value for id: P5
City for given Supplier Smith is London.
Weight of given part is 12.0

PL/SQL procedure successfully completed.

the following Queries: (6)

the details of parts P1, P2 and P3.

select * from parts where pid in ('P1','P2','P3');

PID	PNAME	COLOR	WEIGHT
P1	Nut	Red	12.0
P2	Bolt	Green	17.0
P3	Screw	Blue	17.0

b) Update the status of suppliers as "metro" who belongs to chennai.

SQL> update suppliers set status='metro' where city='Chennai'

o/p: 1 row updated.

c) find the supplier who supply only (five)^{2} parts.

SQL> select sname from suppliers where sid =(select sid from supply group by sid having count(sid)=2);

o/p:

SNAME
Jones

d) find the part names which were not supplied by any supplier

SQL> select pname from parts where pid not in(select pid from supply);

o/p:

SNAME
Nail

e) find the parts with least weight

SQL> select pname from parts where weight =(select min(weight) from parts);

o/p:

PNAME
Nut
Cam

Create & Compute PL/SQL:

a) Display the names of suppliers who belongs to Guntur.

Program:

```
declare
    cursor s is select sname from suppliers where city='Guntur';
    x s%rowtype;
begin
    open s;
    loop
        fetch s into x;
        exit when s%notfound;
        dbms_output.put_line('Suppliers name:'||x.sname);
    close s;
    end;
```

Output:

Suppliers name: Adams

PL/SQL procedure successfully completed.

b) Create a procedure to display the place of a supplier for a given supplier.

```
Create or replace procedure displayplace(name in sailors.sname%type) as
loc sailors.city%type;
begin
    select city into loc from suppliers where sname=name;
    dbms_output.put_line('Place of Supplier '||name||' is :'||loc);
end;
> exec displayplace('Clark')
```

Place of Supplier Clark is London

PL/SQL procedure successfully completed.

Compute the following Queries: (a)

a) Find the parts with least weight.

```
SQL> select pname from parts where weight = (select min(weight) from parts);
```

O/P:

```
PNAME
Nut
Cam
```

b) Include check constraint on quantity such that quantity > 25.

```
SQL> alter table supply add constraint check (quantity>25);
```

O/p: Table altered.

c) Find the names of parts whose weight is less than the avg weight of part

```
SQL> select pname from parts where weight < (select avg(weight) from parts);
```

O/p:

```
PNAME
Nut
Screw
Cam
Nail
```

d) find the supplier who supply only 2 parts.

```
SQL> select sname from suppliers where sid = (select sid from supply group by
        sid having count(sid)=2);
```

O/p:

```
SNAME
Jones
```

e) find the parts who color is red & weight greater than 40.

```
SQL> select pname from parts where color='Red' and weight>40;
```

O/p: no rows selected.

...e & Compute PL/SQL:

Create a package with one procedure & one function.
Procedure finds the name for a given supplier.
Function returns the weight of part for the given part id.

Package Specification:

```
create or replace package pack as
procedure displayname(id suppliers.sid%type);
function displayweight(id parts.pid%type) return varchar2(5);
end;
/
```

o/p: Package Created.

Package Body:

```
create or replace package body pack is
 procedure displayname(id suppliers.sid%type) as
 name sailors.sname%type;
 begin
      select sname into name from suppliers where sid=id;
      dbms_output.put_line('Supplier Name:'||name);
end displayname;
function displayweight(id parts.pid%type) return varchar2(15) as
 x varchar2(15);
 wt parts.weight%type;
 begin
      select weight into wt from parts where pid=id;
       x:=wt;
       return(x);
end displayweight;
end pack;
/
```

o/p: package body created.

Main Program:

```
declare
      id suppliers.sid%type:='&id';
      id1 parts.pid%type:='&id1';
      res varchar2(3);
begin
      pack.displayname(id);
      res:=pack.displayweight(id1);
      dbms_output.put_line('weight of part is:'||res);

end;
```

Output:

Enter value for id: 530
Enter value for id1: P6
Supplier Name : Blake
Weight of Part is 19.0
PL/SQL procedure successfully completed

Compute the following Queries: (12)

a) find the details of P1 & P3.

SQL> select * from parts where pid in ('P1','P3');

O/P:

PID	PNAME	COLOR	WEIGHT
P1	Nut	Red	12.0
P3	Screw	Blue	17.0

b) Update the status of suppliers as "city". who belong to guntur.

SQL> update suppliers set status = "city" where city = 'Guntur'

O/P: 1 row updated.

c) find the Supplier who supply only five parts.

SQL> select sname from suppliers where sid = (select sid from supply group by sid having count(sid) = 5);

O/P:

SNAME
SMITH

d) Find the parts which were supplied by any supplier

SQL> select pname from parts where pid in (select pid from supply);

O/P:

PNAME
Nut
Bolt
Screw
Cam
Cog

e) find the parts with weight greater than 100 and less than 500.

SQL> select pname from parts where weight > 100 and weight < 500.

O/P:

PNAME
Nut

Create SQL:
the names of suppliers who belongs to Guntur.

```
cursor s is select sname from suppliers where city='Guntur';
x s%rowtype;
begin
    open s;
    loop
        fetch s into x;
        exit when s%notfound;
        dbms_output.put_line('Supplier Name:'||x.sname);
    close s;
end;
```

Output:

```
Supplier Name: Adams
PL/SQL procedure successfully completed.
```

b) Create a procedure to display the pname of part for a given pid.

Program:

```
create or replace procedure displayname(id parts.pid%type) as
name parts.pname%type;
begin
    select pname into name from parts where pid=id;
    dbms_output.put_line('Name of the part is:'||name);
end;
```

Output:

```
exec displayname('P4')
Name of the part is: Screw.
PL/SQL procedure successfully completed.
```