*1.find the distinct names and ages of sailors?*

* **SQL> select distinct s.sname ,s.age from sailors s;**

*2.**find all the sailors with rating above 7?*

* **SQL> select s.sname from sailors s where s.rating>7;**

*3.select details of the sailors who have reserved boat no 103?*

* **SQL> select s.\* from sailors s,reserves r where r.bid=103 and s.sid=r.sid;**

*4.find the sids of sailors who have reserved red colour boat?*

* **SQL> select s.sid from sailors s,boats b,reserves r where s.sid=r.sid and b.color='red' and b.bid=r.bid;**

*5.find colours and names of boats reserved by lubber?*

* **SQL> select b.color,b.bname from boats b,reserves r,sailors s where s.sname='lubber'and s.sid=r.sid and r.bid=b.bid;**

*6.find the distinct names of sailors who have reserved atleast one boat?*

* **SQL> select distinct s.sname from sailors s ,reserves r where s.sid=r.sid;**

*7.compute increments for the rating for the person who have sailed two different boats on same day?*

* **SQL> select s.sname,s.rating+1 inc from sailors s,reserves r1,reserves r2 where s.sid=r1.sid and s.sid=r2.sid and r1.bid!=r2.bid and r1.day=r2.day;**

*8.find names and ages of sailors whose name begins and ends with the letter b and has atleast 3 characters?*

* **SQL> select s.sname,s.age from sailors s where s.sname like 'b\_%b';**

*9.list the sailors names, ratings whose age is greater than 35?*

* **SQL> select s.sname,s.rating from sailors s where s.age>35;**

*10.find the sailors names whose rating>8 and age<35?*

* **SQL> select s.sname from sailors s where s.age <35 and s.rating>8;**

**Set Operations:**

*1.find the sids of sailors who have reserved red or a green boat?*

* **SQL> select s.sid from sailors s ,reserves r,boats b where s.sid=r.sid and r.bid=b.bid and b.color='red' union select s1.sid from sailors s1,reserves r1,boats b1 where s1.sid=r1.sid and r1.bid=b1.bid and b1.color='green';**

*2.find the sids of sailors who have reserved both red and green boat?*

* **SQL> select s.sid from sailors s ,reserves r,boats b where s.sid=r.sid and r.bid=b.bid and b.color='red' intersect select s1.sid from sailors s1,reserves r1,boats b1 where s1.sid=r1.sid and r1.bid=b1.bid and b1.color='green';**

*3.find the sids of sailors who have reserved a red boat but not a green boat?*

* **SQL> select s.sid from sailors s ,reserves r,boats b where s.sid=r.sid and r.bid=b.bid and b.color='red' minus select s1.sid from sailors s1,reserves r1,boats b1 where s1.sid=r1.sid and r1.bid=b1.bid and b1.color='green';**

*4.find the sids of sailors who have rating equal to 10 or who have reserved boat no 104?*

* **SQL> select s.sid from sailors s where s.rating=10 union select r1.sid from sailors s1,reserves r1 where s1.sid=r1.sid and r1.bid=104;**

**set compartions operations and nested queries:**

*1.find the sids of sailors whose boat no is 103?*

* **SQL> select s.sid from sailors s where s.sid in(select r.sid from reserves r where r.bid=103);**

*2.find the names of sailors who have reserved boat no 101?*

* **SQL> select s.sname from sailors s where s.sid in(select r.sid from reserves r where r.bid=101);**

*3.find the names of sailors who have not reserved red colour boat?*

* **SQL> select s.sname from sailors s where s.sid in(select r.bid from reserves r where r.bid not in(select b.bid from boats b where b.color='red'));**

*4.find sailors names and ids whose rating is better than some sailors called horatio?*

* **SQL> select s.sname,s.sid from sailors s where s.rating>any(select s1.rating from sailors s1 where s1.sname ='horatio');**

*5.find the sailors with highest rating?*

* **SQL> select s.\* from sailors s where s.rating >=all(select s1.rating from sailors s1);**
* **corelated nested queries:**

*1.find sids from sailors who reserved boat no 103?*

* **SQL> select s.sid from sailors s where exists(select \* from reserves r where r.sid=s.sid and r.bid=103);**

*2.find the names of sialors who have reserved all boats ?*

* **SQL> select s.sid,s.sname from sailors s where not exists((select b.bid from boats b)minus(select r.bid from reserves r where r.sid=s.sid));**
* **queries using aggregate functions:**

*1.find the average of age of all the sailors?*

* **SQL> select avg(s.age) from sailors s;**

*2.find the average of age of sailors with rating 10?*

* **SQL> select avg(s.age) from sailors s where s.rating=10;**

*3.find the name and age of the oldest sailor?*

* **SQL> select s.sname ,s.age from sailors s where s.age=(select max(s1.age) from sailors s1);**

*4.count the no of sailors in sailors relation?*

* SQL> select count(\*) from sailors s;

*5.count the no of distinct sailor names?*

* **SQL> select count(distinct(s.sname)) from sailors s;**

COUNT(DISTINCT(S.SNAME))

*6.find the names of sailor who are older than oldest sailors with rating of 10?*

* **SQL> select s.sname from sailors s where s.age>(select max(s1.age) from sailors s1 where s1.rating=10);**

*7.find the max age from sailors relation?*

* **SQL> select max(s.age) from sailors s;**
* **queries using groupby and having clause**:

*1.find the age of the youngest sailor for each rating level?*

* **SQL> select min(s.age),s.rating from sailors s group by s.rating;**

*2.find the age of the youngest sailor who is eligible to vote for each rating level with atleast 2 sailors?*

* **SQL> select min(s.age),s.rating from sailors s where s.age>=18 group by s.rating having count(s.rating)>=2;**

**PL/SQL**

Write a PL/SQL program Perform Arithmetic operations

**Write a Pl/SQL program to find given number is even or odd**

DECLARE

num number(5);

rem number;

BEGIN

num:=:num;

rem:=mod(num,2);

if rem=0

then

dbms\_output.put\_line(' Number '||num||' is Even');

else

dbms\_output.put\_line(' Number '||num||' is Odd');

end if;

END;

**MULTIPLICATION TABLE**

declare

i number(2);

n number(2);

begin

n:=:n;

for i in 1..10 loop

dbms\_output.put\_line( n || ' \* ' || i || ' = ' || n\*i);

end loop;

end;

**Print 1 to n prime numbers**

declare

i integer;

n integer;

c integer;

begin

n:=:n;

for i in 2..n loop

c:=0;

for j in 1..i loop

if(mod(i,j)=0) then

c:=c+1;

end if;

end loop;

if (c=2) then

dbms\_output.put\_line(i);

end if;

end loop;

end;

Armstrong Number

a)To write a trigger that inserts or updates values of ename and job as uppercase strings even if we give lowercase strings.

CREATE OR REPLACE TRIGGER "EMPT1"

BEFORE

insert or update on "EMP"

for each row

begin

:new.ename:=upper(:new.ename);

:new.job:=upper(:new.job);

end;