1. Create a trigger on account table to maintain a minimum balance Rs. 1000/- in an account.

Step-1: First, create table for "Account" with the following attributes.

Account (acc_no varchar(10) PK, c_name varchar(10), balance number(8,2));

Step-2: Create trigger on account table as "min_bal".

```
create or replace trigger min_bal
before insert or update on account
for each row
declare n number(8,2);
begin
if inserting then
  if(:new.balance<1000) then
    raise_application_error(-20000,'can not create account');
  end if;
elsif updating then
  if(:new.balance<1000) then
    raise_application_error(-20001,'withdrawl not allowed');
  end if;
end if;
end;
/
```

2. Create a trigger on students table to keep a_students table up to date.

Step-1: Create table for students with following attributes.

Students(sid char(10) PK, sname varchar(10), cgpa number(4,2), dob date);

Step-2: Create a copy of students as a_ students with sid, cgpa fields.

Step-3: Create trigger on students as "update_cgpa".

```
create or replace trigger update_cgpa
after insert or delete or update on student_1
for each row
begin
if inserting then
  if(:new.cgpa>=8.0) then
    insert into a_student values(:new.sid,:new.cgpa);
  end if;
elsif deleting then
  if(:old.cgpa>=8.0) then
    delete from a_student where sid=:old.sid;
  end if;
```

```
elsif updating then
  if(:new.cgpa>=8.0) then
    update a_student set cgpa=:new.cgpa where sid=:old.sid;
  end if;
end if;
end;
/
```

3. Create a trigger on reserves table to restrict sailors with rating less than 8 from reserving "Interlake" boats.

Step-1: Create trigger on reserves as "restrict_reserves".

```
create or replace trigger restrict_reserves
after insert on reserves
for each row
declare
n boats.bname%type;
r sailors.rating%type;
begin
  select rating into r from sailors where sid=:new.sid;
  select bname into n from boats where bid=:new.bid;
  if(r<8 and n='interlake') then
    raise_application_error(-20000,'sailor has low rating');
  end if;
end;
/
```

Step-2: To connect server for accessing existed tables, execute the following command

```
Set serveroutput on;
```

4. Create a trigger on sailors table to display count of sailors in sailors table when a new row is inserted.

Step-1: Create trigger on sailors as "count_sailors".

```
create or replace trigger count_sailors
after insert on sailors
declare
n number(5);
begin
  select count(*) into n from sailors;
  dbms_output.put_line('There are'||n||'sailors');
end;
/
```

Step-2: To connect server for accessing existed tables, execute the following command. Set serveroutput on;

5. Create a trigger on employees table which raises a warning message when the total salary paid to all employees together in department 10 exceeds 100000, department 20 exceeds 60000 and department 30 exceeds 95000;

Step-1: Create a trigger on employees table as "total_sal_limit".

```
create or replace trigger total_sal_limit
after insert or update on emp
declare
n number(8,2);
m number(8,2);
p number(8,2);
begin
  select sum(sal) into m from emp where did=10;
  select sum(sal) into n from emp where did=20;
  select sum(sal) into p from emp where did=30;
  if(m>100000 or n>60000 or p>95000) then
    raise_application_error(-20000,'salary exceeding budget');
  end if;
end;
/
```

Step-2: To connect server for accessing existed tables, execute the following command.

```
Set serveroutput on;
```

Department table

(10, 'ACCOUNTING', 'NEW YORK');
(20, 'RESEARCH', 'DALLAS');
(30, 'SALES', 'CHICAGO');
(40, 'OPERATIONS', 'BOSTON');
Commit;

Data of Salary_Grade Table:

('PRESIDENT',25000,50000);
('MANAGER',15000,23000);
('CLERK',10000,15000);
('SALESMAN',10000,15000);
('ANALYST',13000,20000);
Commit;

Data of Employees Table:

(7839, 'KING', '12-Aug-1980','PRESIDENT', null,'17-Nov-2020',30000, null, 10);
(7698, 'BLAKE', '11-Jul-1990','MANAGER', 7839, '1-May-2020',17000, null, 30);
(7782, 'CLARK', 24-Dec-1989','MANAGER', 7839, '9-Jun-2020',16500, null, 10);
(7566, 'JONES', '17-Oct-1990','MANAGER', 7839, '2-Apr-2020',17000, null, 20);
(7788, 'SCOTT', '22-May-1993','ANALYST', 7782, '13-Jul-2020',15000, null, 10);
(7902, 'FORD', '9-Apr-1992','ANALYST', 7566, '3-Dec-2020', 17000, null, 20);
(7369, 'SMITH', '2-Feb-1993','CLERK', 7566, '17-Dec-2019', 10000, null, 20);
(7499, 'ALLEN', '13-Jan-1993', 'SALESMAN', 7698,'20-Feb-2020',12000, 1300, 30);
(7521, 'WARD', '21-Apr-1992','SALESMAN', 7698,'22-Feb-2020',12500, 1500, 30);
(7654, 'MARTIN', '9-May-1993','SALESMAN', 7698,'28-Sep-2020',12500, 1400, 30);
(7844, 'TURNER', '11-Jun-1994','SALESMAN', 7698,'8-Sep-2020', 15000, 0, 30);
(7876, 'ADAMS', '25-Sep-1994', 'CLERK', 7566,'13-JUL-2017',11000, null, 20);
(7900, 'JAMES', '23-Nov-1993', 'CLERK', 7698, '3-12-2020', 11000, null, 30);
(7934, 'MILLER', '18-Jun-1993','CLERK', 7782, '23-1-2017', 12000, null, 10);
Commit;

1. Display IDs and names of employees whose names begin with character 'A'.
SQL> select emp_id,ename from emp where ename like 'A%';

O/P:

7499 ALLEN

7876 ADAMS

---

2. Display IDs and names of employees whose names are 4 characters long.
SQL>select emp_id,ename from emp where ename like '____';

O/P:

7839 KING

7902 FORD

7521 WARD

---

3. Display all the information of the EMP table?
SQL>select * from emp;

O/P:

7839 KING PRESIDENT 12-08-80 17-11-20 30000
10

7698	BLAKE	MANAGER	7839	11-07-90	01-05-20	17000	30
7782	CLARK	MANAGER	7839	24-12-89	09-06-20	16500	10
7566	JONES	MANAGER	7839	17-10-90	02-04-20	17000	20
7788	SCOTT	ANALYST	7782	22-05-93	13-07-20	15000	10
7902	FORD	ANALYST	7566	09-04-92	03-12-20	17000	20
7369	SMITH	CLERK	7566	02-02-93	17-12-19	10000	20
7499	ALLEN	SALESMAN	7698	13-01-93	20-02-20	12000	1300 30
7521	WARD	SALESMAN	7698	21-04-92	22-02-20	12500	1500 30
7654	MARTIN	SALESMAN	7698	09-05-93	28-09-20	12500	1400 30
7844	TURNER	SALESMAN	7698	11-06-94	08-09-20	15000	0 30
7876	ADAMS	CLERK	7566	25-09-94	13-07-17	11000	20
7900	JAMES	CLERK	7698	23-11-93	03-12-20	11000	30

7934 MILLER CLERK 7782 18-06-93 23-01-17 12000 10

---

4. Display unique Jobs from EMP table?

SQL>select distinct job from emp;

O/P:

CLERK

SALESMAN

PRESIDENT

MANAGER

ANALYST

---

5. Display ID of each employee and name of his manager.
SQL>select e1.emp_id,e2.ename from emp e1,emp e2 where e2.emp_id=e1.mgr_id;

O/P:

7876 JONES

7369 JONES

7902 JONES

7499 BLAKE

7900 BLAKE

7844 BLAKE

7654 BLAKE

7521 BLAKE

7934 CLARK

7788 CLARK

7698 KING

7782 KING

7566 KING

---

6. List the details of employees in the ascending order of their Salaries?
SQL>select * from emp order by sal asc;

O/P:

7369	**SMITH**	**CLERK**	**7566**	**02-02-93**	**17-12-19**	**10000**
	20					
7900	**JAMES**	**CLERK**	**7698**	**23-11-93**	**03-12-20**	**11000**
	30					
7876	**ADAMS**	**CLERK**	**7566**	**25-09-94**	**13-07-17**	**11000**
	20					

7499 ALLEN SALESMAN 7698 13-01-93 20-02-20 12000 1300 30

7934 MILLER CLERK 7782 18-06-93 23-01-17 12000 10

7654 MARTIN SALESMAN 7698 09-05-93 28-09-20 12500 1400 30

7521 WARD SALESMAN 7698 21-04-92 22-02-20 12500 1500 30

7788 SCOTT ANALYST 7782 22-05-93 13-07-20 15000 10

7844 TURNER SALESMAN 7698 11-06-94 08-09-20 15000 0 30

7782 CLARK MANAGER 7839 24-12-89 09-06-20 16500 10

7566 JONES MANAGER 7839 17-10-90 02-04-20 17000 20

7698 BLAKE MANAGER 7839 11-07-90 01-05-20 17000 30

---

7. Display ID of each employee and location of his working department.

SQL>select e.emp_id,d.location from emp e,dept d where e.DEPT_ID=d.DEPT_ID;

O/P: 7934 NEW YORK

7788 NEW YORK

7782 NEW YORK

7839 NEW YORK

7902 DALLAS

7566 DALLAS

7369 DALLAS

7876 DALLAS

7521 CHICAGO

7654 CHICAGO

7844 CHICAGO

7900 CHICAGO

7499 CHICAGO

7698 CHICAGO

---

8. List the employees who joined before 2020.
SQL>select ename from emp where hire_date <'1-1-2020';

O/P:

SMITH

ADAMS

MILLER

---

9. List the Emp_id, Ename, and Salary of all employees working for Mgr 7698.
SQL>select emp_id,ename,sal from emp where MGR_ID=7698;

O/P:

7499 ALLEN 12000

7521 WARD 12500

7654 MARTIN 12500

7844 TURNER 15000

7900 JAMES 11000

---

10. Display the details of employees who do not have allowance.
SQL>select * from emp where ALNC is null;

O/P:

7839 KING PRESIDENT 12-08-80 17-11-20 30000
10

7698 BLAKE MANAGER 7839 11-07-90 01-05-20 17000
30

7782	CLARK	MANAGER	7839	24-12-89	09-06-20	16500
	10					
7566	JONES	MANAGER	7839	17-10-90	02-04-20	17000
	20					
7788	SCOTT	ANALYST	7782	22-05-93	13-07-20	15000
	10					
7902	FORD	ANALYST	7566	09-04-92	03-12-20	17000
	20					
7369	SMITH	CLERK	7566	02-02-93	17-12-19	10000
	20					
7876	ADAMS	CLERK	7566	25-09-94	13-07-17	11000
	20					
7900	JAMES	CLERK	7698	23-11-93	03-12-20	11000
	30					
7934	MILLER	CLERK	7782	18-06-93	23-01-17	12000
	10					

---

11. Display all the details of the employees whose allowance is more than their salary.

SQL>select * from emp where ALNC>sal;

O/P:

---

12. List the employees who are either ‘CLERK’ or ‘ANALYST’ in the descending order.
SQL>select * from emp where job='CLERK' or job='ANALYST' order by job desc;

O/P:

7369	**SMITH**	**CLERK**	**7566**	**02-02-93**	**17-12-19**	**10000**
	20					
7900	**JAMES**	**CLERK**	**7698**	**23-11-93**	**03-12-20**	**11000**
	30					
7934	**MILLER**	**CLERK**	**7782**	**18-06-93**	**23-01-17**	**12000**
	10					
7876	**ADAMS**	**CLERK**	**7566**	**25-09-94**	**13-07-17**	**11000**
	20					
7902	**FORD**	**ANALYST**	**7566**	**09-04-92**	**03-12-20**	**17000**
	20					
7788	**SCOTT**	**ANALYST**	**7782**	**22-05-93**	**13-07-20**	**15000**
	10					

---

13. List the employees who are working for the dept_id 10 or 20.
SQL>select * from emp where dept_id=10 OR dept_id=20;

O/P:

7839	KING	PRESIDENT		12-08-80	17-11-20	30000
	10					
7782	CLARK	MANAGER	7839	24-12-89	09-06-20	16500
	10					
7566	JONES	MANAGER	7839	17-10-90	02-04-20	17000
	20					
7788	SCOTT	ANALYST	7782	22-05-93	13-07-20	15000
	10					
7902	FORD	ANALYST	7566	09-04-92	03-12-20	17000
	20					
7369	SMITH	CLERK	7566	02-02-93	17-12-19	10000
	20					
7876	ADAMS	CLERK	7566	25-09-94	13-07-17	11000
	20					
7934	MILLER	CLERK	7782	18-06-93	23-01-17	12000
	10					

---

14. List the employees those are having four chars and third character must be ‘r’.

SQL>select * from emp where length(ename)=4 and ename like '__R%';

O/P:

7902 FORD ANALYST 7566 09-04-92 03-12-20 17000 20

7521 WARD SALESMAN 7698 21-04-92 22-02-20 12500 1500 30

---

15. List the employees whose allowance is four digit number ending with Zero.
SQL>select * from emp where length(alnc)=4 and alnc like '___0';

O/P:

7499 ALLEN SALESMAN 7698 13-01-93 20-02-20 12000 1300 30

7521 WARD SALESMAN 7698 21-04-92 22-02-20 12500 1500 30

7654 MARTIN SALESMAN 7698 09-05-93 28-09-20 12500 1400 30

---

16. List the employee who does not belong to dept_no 20.
SQL>select * from emp where dept_id<>20;

O/P:

7839 KING PRESIDENT 12-08-80 17-11-20 30000 10

7698 BLAKE MANAGER 7839 11-07-90 01-05-20 17000 30

7782 CLARK MANAGER 7839 24-12-89 09-06-20 16500 10

7788 SCOTT ANALYST 7782 22-05-93 13-07-20 15000 10

7499 ALLEN SALESMAN 7698 13-01-93 20-02-20 12000 1300 30

7521 WARD SALESMAN 7698 21-04-92 22-02-20 12500 1500 30

7654 MARTIN SALESMAN 7698 09-05-93 28-09-20 12500 1400 30

7844 TURNER SALESMAN 7698 11-06-94 08-09-20 15000 0 30

7900 JAMES CLERK 7698 23-11-93 03-12-20 11000 30

7934 MILLER CLERK 7782 18-06-93 23-01-17 12000 10

---

17. Display IDs and completed years of work experience of all employees.

SQL>select emp_id,round(months_between(sysdate,hire_date)/12) as exp from emp;

O/P:

7839 1

7698 2

7782 1

7566 2

7788 1

7902 1

7369 2

7499 2

7521 2

7654 1

7844 1

7876 4

7900 1

7934 5

---

18. Display IDs and hire dates of employees joined between the years 2020 and 2018.
SQL>select emp_id,hire_date from emp where to_char(hire_date,'yyyy') between 2018 and 2020;

O/P:

7839 17-11-20

7698 01-05-20

7782 09-06-20

7566 02-04-20

7788 13-07-20

7902 03-12-20

7369 17-12-19

7499 20-02-20

7521 22-02-20

7654 28-09-20

7844 08-09-20

7900 03-12-20

---

19. Display IDs and hire dates of employees who joined in any month between July and October.
SQL>select emp_id,hire_date from emp where to_char(hire_date,'mm') between 07 and 10;

O/P:

7788 13-07-20

7654 28-09-20

7844 08-09-20

7876 13-07-17

---

20. Display maximum and minimum salaries of jobs from the sal_range table.
SQL>select max_sal as maximum,min_sal as minimum,job from sal_range;

O/P:

50000	**25000**	**PRESIDENT**
23000	**15000**	**MANAGER**
15000	**10000**	**CLERK**
15000	**10000**	**SALESMAN**
20000	**13000**	**ANALYST**

21. Display first four rows of employee table.
SQL>select * from emp where rownum<5;

O/P:

7839 KING PRESIDENT 12-08-80 17-11-20 30000
10

7698 BLAKE MANAGER 7839 11-07-90 01-05-20 17000
30

7782 CLARK MANAGER 7839 24-12-89 09-06-20 16500
10

7566 JONES MANAGER 7839 17-10-90 02-04-20 17000
20

---

22. Display maximum and minimum salaries and the corresponding jobs of emp table.

```
SQL>select job,sal from emp where sal>=all(select max(sal) from emp)
union
select job,sal from emp where sal<=all(select min(sal) from emp);
                                or
select e1.job , e1.sal from emp e1 where e1.sal=(select max(sal) from
emp e2) or e1.sal=(select min(sal) from emp e2);
```

O/P:

```
CLERK       10000
PRESIDENT       30000
```

23. Display the count of employees working under each manager along with ID of manager.

```
Sql>select mgr_id,count(emp_id) as count  from emp where mgr_id is
not null group by mgr_id;
```

o/p:

```
7839 3
7782 2
7698 5
7566 3
```

24. Obtain total salary of all employees for each job title and display job title having highest total salary.

Sql: select job,sum(sal) from emp group by job having sum(sal)>=all(select sum(sal) from emp group by job);

o/p:

SALESMAN 52000

25. Display each location and the total salary paid at each location.

Sql>select d.location, sum(sal) from dept d,emp e where e.dept_id=d.dept_id group by d.location;

o/p:

NEW YORK 73500

CHICAGO 80000

DALLAS 55000

26. Display the details of employees working in departments 10 and 20 in the ascending order of department numbers.

Sql>select * from emp where dept_id=10 or dept_id=20 order by dept_id;

o/p:

7839	KING	PRESIDENT		12-08-80	17-11-20	30000	10
7782	CLARK	MANAGER	7839	24-12-89	09-06-20	16500	10
7788	SCOTT	ANALYST	7782	22-05-93	13-07-20	15000	10
7934	MILLER	CLERK	7782	18-06-93	23-01-17	12000	10
7902	FORD	ANALYST	7566	09-04-92	03-12-20	17000	20
7369	SMITH	CLERK	7566	02-02-93	17-12-19	10000	20
7876	ADAMS	CLERK	7566	25-09-94	13-07-17	11000	20
7566	JONES	MANAGER	7839	17-10-90	02-04-20	17000	20

27. Display IDs and names of employees from 2nd row to 5th row of Employee table.

```
Sql>select rn,* from(select rownum as rn,* from emp) where rn between 2 and 5;
Or
select emp_id,ename from emp where rownum<=5

minus
select emp_id,ename from emp where rownum<2;
or
select rn,emp_id,ename from(select rownum as rn,emp_id,ename from emp) where rn between 2 and 5;
o/p:
2       7698 BLAKE
3       7782 CLARK
4       7566 JONES
5       7788 SCOTT
```

28. Display IDs and names of employees of only 2nd row.

```
Sql>select rn,emp_id,ename from (select rownum as rn,e.* from emp e) where rn=2;
o/p: 2       7698 BLAKE
```

29. Display last four rows of employee table.

```
Sql>select  * from(select rownum as rn,e.* from emp e order by rn desc) where rownum<=4;
Or
select emp_id,ename from emp where rownum<=14
minus
select emp_id,ename from emp where rownum<=10;
o/p:
14      7934 MILLER     CLERK      7782 18-06-93   23-01-17   12000
        10
13      7900 JAMES      CLERK      7698 23-11-93   03-12-20   11000
        30
```

12 7876 ADAMS CLERK 7566 25-09-94 13-07-17 11000 20

11 7844 TURNER SALESMAN 7698 11-06-94 08-09-20 15000 0 30

30. Display IDs and names of employees in all even rows.

Sql>select rn,emp_id,ename from (select rownum as rn,e.* from emp e) where mod(rn,2)=0;

o/p:

2 7698 BLAKE

4 7566 JONES

6 7902 FORD

8 7499 ALLEN

10 7654 MARTIN

12 7876 ADAMS

14 7934 MILLER

31. Determine the age of each employee at the time of hiring and display ID, date of birth and hire date of employee who joined the company at highest age among all.

Sql>select emp_id,dob,hire_date,age from(select e.*,round(months_between(hire_date,dob)/12) as age from emp e) where age>=all(select max(age) from(select e.*,round(months_between(hire_date,dob)/12) as age from emp e));

o/p:

7839 12-08-80 17-11-20 40

Exercise 3: Creation and Querying of Sailors database. This database contains **three** tables.
i) **Sailors** which describe a sailor using **Sailor ID**, **Sailor Name**, **Age** and **Rating Level**.
ii) **Boats** which describe a boat using **Boat ID**, **Boat Name** and **Colour**.
iii) **Reserves** which describe a reservation using **Sailor ID** which is a foreign key referencing Sailors table, **Boat ID** which is a foreign key referencing Boats table and the **Date of Reservation**.

SID	SNAME	RATING	AGE
22	Dustin	7	45
29	Brutus	3	33
31	Lubber	9	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	2	18
74	Horatio	9	35
85	Art	3	25
95	Bob	8	63
100	Horatio	7	45

Sailors Table Instance

BID	BNAME	COLOR
101	Interlake	Blue
102	Interlake	Red
103	Clipper	Green
104	Marine	Red
105	Marine	Red

Boats Table Instance

SID	BID	DAY
22	101	10-OCT-2020
22	102	10-OCT-2020
22	103	10-AUG-2020
22	104	10-JUL-2020
31	102	11-OCT-2020
31	103	11-JUN-2020
31	104	11-DEC-2020
64	101	09-MAY-2020
64	102	09-AUG-2020
74	103	09-AUG-2020
95	103	10-JUN-2020
58	102	11-OCT-2020
58	104	11-DEC-2020
100	102	15-JUN-2020
31	101	22-NOV-2020
22	105	20-OCT-2020
31	105	22-NOV-2020
74	101	23-MAY-2020

Reserves table Instance

1)Display names of sailors that sailed a boat on 10-oct-2020.
Sql>select distinct s.s_name from sailors s,reserves r where s.s_id=r.s_id and r.d_o_r='10-10-2020';
o/p:
DUSTIN
2)Display colors of boats sailed by sailor 64.
Sql>select b.b_color from boats b,reserves r where b.b_id=r.b_id and r.s_id=64;

o/p:
BLUE
RED

3)Display color of boats sailed by sailor with highest age.
Sql>select b.b_color from boats b,sailors s,reserves r where s.s_age=(select max(s_age) from sailors) and b.b_id=r.b_id and s.s_id=r.s_id;
o/p:
GREEN

4)For each reservation made display sailor name and boat name.
Sql>select s.s_name,b.b_name from sailors s,boats b,reserves r where r.s_id=s.s_id and r.b_id=b.b_id;
o/p:
DUSTIN MARINE
DUSTIN MARINE
DUSTIN CLIPPER
DUSTIN INTERLAKE
DUSTIN INTERLAKE
LUBBER MARINE
LUBBER MARINE
LUBBER CLIPPER
LUBBER INTERLAKE
LUBBER INTERLAKE
RUSTY MARINE
RUSTY INTERLAKE
HORATIO INTERLAKE
HORATIO INTERLAKE
HORATIO CLIPPER
HORATIO INTERLAKE
BOB CLIPPER
HORATIO INTERLAKE

5)Display colors of boats sailed by sailors with rating>average rating of sailors with age
above 25 years.
Sql>select b.b_color from boats b,sailors s,reserves r where s.sr_l>(select avg(sr_l) from sailors) and s.s_age>25 and b.b_id=r.b_id and s.s_id=r.s_id;
o/p:
GREEN
RED
BLUE
RED
RED

GREEN
RED
BLUE
RED
RED
RED
RED
BLUE
RED
BLUE
GREEN
GREEN
RED

6)Display count of sailors with rating above 7 and age above 25.
Sql>select count(s_id) as count from sailors where sr_l>7 and s_age>25;
o/p:
4

7)Display information of reservation made by any sailor "Horatio".
Sql>select r.* from sailors s,reserves r where s.s_name like 'HORATIO' and s.s_id=r.s_id;
o/p:

64	101	09-05-20
64	102	09-08-20
74	101	23-05-20
74	103	09-08-20
100	102	15-06-20

8)Display id(s) of sailors who made minimum number of reservations.
Sql>select s_id,COUNT(s_id) from reserves group by s_id having count(s_id)<=all(select count(s_id) from reserves group by s_id);
o/p:

95	1
100	1

9)Display names of sailors who made at least two reservations on the same day.
Sql>select distinct s.s_id,s.s_name,r1.d_o_r from sailors s,reserves r1,reserves r2 where s.s_id=r1.s_id and r1.s_id=r2.s_id and r1.d_o_r=r2.d_o_r and r1.b_id!=r2.b_id;
o/p:

22	DUSTIN	10-10-20
31	LUBBER	22-11-20

10)a.Display id of sailors who reserved all boats.

```
Sql>select s_id,COUNT(s_id) from reserves group by s_id having count(s_id)>=all(select count(s_id) from
reserves group by s_id);
o/p:
22      5
31      5
b.sql> select s.s_id,s.s_name from sailors s where not exists((select b_id from boats)
minus
(select r.b_id from reserves r where r.s_id=s.s_id));
o/p:
22      DUSTIN
31      LUBBER
```

11)Display id of sailors that reserved all red boats.

```
Sql>select s.s_id,s.s_name from sailors s where not exists((select b_id from boats where b_color='RED')
minus
(select r.b_id from reserves r where r.s_id=s.s_id));
o/p:
22      DUSTIN
31      LUBBER
```

12)Display id's of sailors who have reseved both Red and Blue boats.

```
Sql>select r.s_id from reserves r,boats b where r.b_id=b.b_id and b.b_color='RED'
intersect
select r.s_id from reserves r,boats b where r.b_id=b.b_id and b.b_color='BLUE';
o/p:
22
31
64
```

13)Display ids of sailors that reserved atleast two different color boats.

```
Sql>select r.s_id from reserves r,boats b where r.b_id=b.b_id group by r.s_id having count(distinct
b.b_color)>=2;
o/p:
22
31
74
64
```

14)Display names of sailors as pairs for each rating that has two sailors.

Sql>select s1.sr_l,s1.s_name,s2.s_name from sailors s1,sailors s2 where s1.s_id<>s2.s_id and s1.sr_l=s2.sr_l and s1.s_name<s2.s_name;

o/p:

3 ART BRUTUS

9 HORATIO LUBBER

7 DUSTIN HORATIO

8 ANDY BOB

7 DUSTIN HORATIO

15)Display id,name of sailors having third highest rating.

sql>select s.s_id,s.s_name from sailors s where s.sr_l=(select sr_l from
(select rownum as rn,sr_l from (select distinct sr_l from sailors order by sr_l desc)) where rn=3);

o/p:

32 ANDY

95 BOB

16)Diplay names of sailors that have not reserved "Interlake" boats.

Sql>select s.s_id,s.s_name from sailors s,boats b,reserves r where r.s_id=s.s_id and r.b_id=b.b_id and s.s_id not in
(select r.s_id from reserves r,boats b where r.b_id=b.b_id and b.b_name='INTERLAKE');

o/p:

95 BOB

17) Find the days on which any two sailors named " Horatio ", each made a reservation.

Sql>select s.s_id,s.s_name,r1.d_o_r from sailors s,reserves r1,reserves r2 where s.s_id=r1.s_id and s.s_name='HORATIO' and r1.s_id<>r2.s_id and r1.d_o_r=r2.d_o_r;

o/p:

74 HORATIO 09-08-20

64 HORATIO 09-08-20