DBMS LAB ON 12/02/2019

Sailors(sid: integer, sname: string, rating: integer, age: real);
Boats(bid: integer, bname: string, color: string);
Reserves(sid: integer, bid: integer, day: date).

Sailors

Sid	Sname	Rating	Age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	40
85	Art	3	25.5
95	Bob	3	63.5

Boats

bid	bname	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

Reserves

sid	bid	day
22	101	1998-10-10
22	102	1998-10-10
22	103	1998-10-8
22	104	1998-10-7
31	102	1998-11-10
31	103	1998-11-6
31	104	1998-11-12
64	101	1998-9-5
64	102	1998-9-8
74	103	1998-9-8

Figure 1: Instances of Sailors, Boats and Reserves

1. Create the Tables:

```
CREATE TABLE sailors (      sid integer not null,
                            sname varchar(32),
                            rating integer,
                            age real,
                            CONSTRAINT PK_sailors PRIMARY KEY (sid) );
CREATE TABLE reserves (     sid integer not null,
                            bid integer not null,
                            day datetime not null,
                            CONSTRAINT PK_reserves PRIMARY KEY (sid, bid, day),
                            FOREIGN KEY (sid) REFERENCES sailors(sid),
                            FOREIGN KEY (bid) REFERENCES boats(bid) );
```

Queries:

1. Find all information of sailors who have reserved boat number 101.

```
SELECT S.*
FROM Sailors S, Reserves R
WHERE S.sid = R.sid AND R.bid = 101
```

2. Find the names of sailors who have reserved a red boat, and list in the order of age.

```
SELECT S.sname, S.age
FROM Sailors S, Reserves R, Boats B
WHERE S.sid = R.sid AND R.bid = B.bid AND B.color = 'red'
ORDER BY S.age
```

3. Find the names of sailors who have reserved at least one boat.

```
SELECT sname
FROM Sailors S, Reserves R
WHERE S.sid = R.sid
```

4. Find the ids and names of sailors who have reserved two different boats on the same day.

```
SELECT DISTINCT S.sid, S.sname
FROM Sailors S, Reserves R1, Reserves R2
WHERE S.sid = R1.sid AND S.sid = R2.sid
        AND R1.day = R2.day AND R1.bid <> R2.bid
```

5. Find the name and age of the sailors whose name begins and ends with 'B' and has ateast 3 characters

```
SELECT sname, age, rating + 1 as sth
FROM Sailors
WHERE 2* rating –1 < 10 AND sname like 'B_%b'
```

Union, Intersect and Except

1. Find the ids of sailors who have reserved a red boat or a green boat.

```
SELECT R.sid
FROM Boats B, Reserves R
WHERE R.bid = B.bid AND B.color = 'red'
UNION
SELECT R2.sid
FROM Boats B2, Reserves R2
WHERE R2.bid = B2.bid AND B2.color = 'green'
```

2. Find the names of the sailors who have reserved both a red or a yellow boat.

```
select s.sname
from sailors s, boats b, reserves r
where s.sid=r.sid and b.bid=r.bid
 and (b.color='red' or b.color='yellow');
```

3. Find all sids of sailors who have a rating of 10 or have reserved boat 111.

```
 select s.sid
 from sailors s
where s.rating = 10
union
select r.sid
from reserves r
where r.bid = 111;
```

4. Find all sids of sailors who have reserved red boats but not yellow boats.

```
select s.sid
from sailors s, boats b, reserves r
where s.sid = r.sid and r.bid = b.bid and b.color = 'red'
minus
select s2.sid
from sailors s2, boats b2, reserves r2
where s2.sid = r2.sid and r2.bid = b2.bid and b2.color = 'yellow';
```

5. Find the names of the sailors who have reserved both a red and a yellow boat.

```
select s.sname
 from sailors s, boats b, reserves r
  where s.sid = r.sid and r.bid = b.bid and b.color = 'red'
  intersect
  select s2.sname
  from sailors s2, boats b2, reserves r2
  where s2.sid = r2.sid and r2.bid = b2.bid and b2.color = 'yellow';
```

Nested Query

1. Find the names of sailors who have reserved boat 103.

```
SELECT S.sname
FROM Sailors S
WHERE S.sid IN (    SELECT R.sid
                    FROM Reserves R
                    WHERE R.bid = 103   )
```

2. Find the name and the age of the youngest sailor.

```
SELECT S.sname, S.age
FROM Sailors S
WHERE S.age <= ALL (  SELECT age
                      FROM Sailors )
```

3. Find the names and ratings of sailor whose rating is better than some sailor called Horatio.

```
SELECT S.sname, S.rating
FROM Sailors S
WHERE S.rating > ANY ( SELECT S2.rating
                       FROM Sailors S2
                       WHERE S2.sname = 'Horatio')
```

4. Find the names of sailors who have reserved all boats.

```
SELECT S.sname
FROM Sailors S
WHERE NOT EXISTS (  ( SELECT B.bid
                      FROM Boats B)
                    EXCEPT
                    ( SELECT R.bid
                      FROM Reserves R
                      WHERE R.sid = S.sid      ) )
```

EMPLOYEE TABLE

5. Display all the departments where department has 3 employees?

Select deptno from dept a where deptno in(select deptno from emp group by deptno having count(*)=3)

6. Display all the departments where department does not have any employees

select deptno from dept where not exists(select 1 from emp where emp.deptno = dept.deptno);

select deptno from dept where deptno not in(select deptno from emp);

7. : Display all the departments where department does have atleast one employee

select * from dept a where exists(select 1 from emp b where b.deptno = a.deptno)

select * from dept a where deptno in(select deptno from emp b where a.deptno = b.deptno)

8. Display all the records for deptno which belongs to employee name JAMES?

select * from emp where deptno in(select deptno from emp where ename = 'JAMES')

9. Display all the records in emp table where salary should be less then or equal to ADAMS salary?

select * from emp where sal <= (select sal from emp where ename='ADAMS')

10. Display all employees those were joined before employee WARD joined?

select * from emp where hiredate < (select hiredate from emp where ename='WARD')

11. Display all subordinate those who are working under BLAKE?

Select ename from emp where mgr = (select empno from emp where ename='BLAKE')

12. Display all subordinate(all levels) for employee BLAKE?

select ename from emp start with empno = (select empno from emp where ename='BLAKE') connect by prior empno = mgr

13. Display all record in emp table for deptno which belongs to KING's Job?

select * from emp where deptno in(select deptno from emp where job= (select job from emp where ename = 'KING'))

14. Display the employees for empno which belongs to job PRESIDENT?

select * from emp where empno in(select empno from emp where ename in(select ename from emp where JOB = 'PRESIDENT'));

15. Remove all the employees in SMITH's department

delete emp where deptno = (select deptno from emp where ename = 'SMITH')

16. Update MARTIN salary same as SMITH's salary

update emp set sal = (select sal from emp where ename = 'SMITH') where ename='MARTIN'

17. Display all employees where their salary is less then the Ford’s salary?

select * from emp where sal<(select sal from emp where ename=’FORD’);

18. Display all records in EMP table those were joined before SCOTT joined?

select * from emp where hiredate<(select hiredate from emp where ename=’SCOTT’)

19. Display all ename, sal, deptno,dname from emp, dept table where all department which has employees as well as department does not have any employees. This query should includenon matching rows.

select dname,b.deptno, ename,sal from emp a right outer join dept b on a.deptno = b.deptno;

20. Display all ename, sal, deptno from emp, dept table where all employees which has matching department as well as employee does not have any departments. This query should includenon matching rows.

select dname,b.deptno, ename,sal from emp a left outer join dept b on a.deptno = b.deptno;

21. Display all ename, sal, deptno from emp, dept table where all employees which has matching and non matching department as well as all departments in dept table which has matching and non matching employees. This query should include non matching rows on both the tables.

Select dname,b.deptno, ename,sal from emp a full outer join dept b on a.deptno = b.deptno

22. Create table emp1 and copy the emp table for deptno 10 while creating the table

Create table emp1 as select * from emp where deptno=10

23. Create table emp2 with same structure of emp table. Do not copy the data

create table emp2 as select * from emp where 1=2;

AGGREGATION OPERATORS

1. Count the number of different sailor names

```
SELECT COUNT( DISTINCT S.sname )
FROM Sailors S
```

2. Calculate the average age of all sailors

```
SELECT AVG(s.age)
FROM Sailors S
```

3. Find the name and the age of the youngest sailor.

```
SELECT S.sname, S.age
FROM Sailors S
WHERE S.age = (SELECT MIN(S2.age)
                FROM Sailors S2 )
```

GROUPBY HAVING CLAUSE

1. Find the average age of sailors for each rating level

```
SELECT S.rating, AVG(S.age) AS avg_age
FROM Sailors S
GROUP BY S.rating
```

2. Find the average age of sailors for each rating level that has at least two sailors.

```
SELECT S.rating, AVG(S.age) AS avg_age
FROM Sailors S

GROUP BY S.rating
HAVING COUNT(*) > 1
```

3. An example shows difference between WHERE and HAVING:

```
SELECT S.rating, AVG(S.age) as avg_age
FROM Sailors S
WHERE S.age >=40
GROUP BY S.rating
```