RDBMS Lab Exercise Solutions

SQL Statements

1. Given three relations- sailors, boats and reserves. Sid, Bid and (Sid, Bid) are the primary keys of sailors, boats and reserves respectively. Sid and Bid are also the foreign keys of reserves which references Sid and Bid of sailors and boats relation respectively. No two sailors have same rating.

Sid	Sname	Rating	age
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	9	55
32	Andy	8	25.0
58	Rusty	10	35

sailors

Bid	Bname	Color
101	Interlake	Blue
102	Interlake	Red
103	Clipper	Green
104	Marine	Red

boats

Sid	Bid	Day
22	101	1/1/2004
22	102	1/1/2004
22	103	1/2/2004
31	103	5/5/2005
32	104	7/4/2005

reserves

Write SQL queries for the following:

1. Find the names of sailors who have reserved a red boat.

select sname from sailors where sid in (select sid from reserves where bid in (select bid from boats where color = "red"));

2. Find the names of the Sailors who have reserved at least one boat

select sname from sailors where sid in (select sid from reserves group by sid);

3. Compute increments for the ratings of persons who have sailed two different boats on the same day.

update sailors set rating = rating+5 where sid =(select distinct(r1.sid) from reserves r1, reserves r2 where r1.day = r2.day and r1.bid <> r2.bid);

4. Find the ages of sailors whose name begins and ends with B and has at least 3 characters.

    select age from sailors where sname like 'b_%b';

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

 select sname from sailors where sid in (select sid from reserves where bid in (

    select bid from boats where color = "red" or color = "green"));

6. Find the sids of all sailors who have reserved red boats but not green boats.

select sid from sailors where sid in (select sid from reserves where bid in (

    select bid from boats where bid not in (select bid from boats where color="green" or color ="blue")));

7. Find the sailors with the highest rating

select sname from sailors where rating = (select max(rating) from sailors);

8. Find the name of the oldest sailor.

select sname from sailors where age = (select max(age) from sailors);

9. Count the number of different sailor names.

select count(sname) from sailors group by sname;

10. Find the no. of sailors who is eligible to vote for each rating level.

11. Find the no. of sailors who is eligible to vote for each rating level with at least two such sailors.

12. Find the sid of the sailors who have sailed exactly one boat.

select sid from reserves group by sid having count(sid) = 1;

13. Find sailors who have not reserved any boats.

select sid from sailors where sid not in(select sid from reserves group by sid having count(sid) >= 1);

14. Find the Sailors who have reserved all the boats.

select sid from reserves group by bid having count(bid) > (select count(bid) from boats);

15. Find all the sailors older than Dustin.

select s.sname from sailors s where s.age > (select s1.age from sailors s1 where s1.sname ="Dustin");

16. Find all sailors whose ratings is greater than any others rating without using aggregates like MAX.

select sname from sailors order by rating desc limit 0, 1;

(select sname from sailors order by rating desc limit n-1, 1;)

17. Find the sailors with 3rd highest ratings.

select sname from sailors order by rating desc limit 2, 1;

18. Find sids of the sailors who have reserved all the boats reserved by the sailor with sid ='31'.

select sid from reserves where bid = (select bid from reserves where sid = 31);

19. List out all the sailors. For the sailors who have reserved some boats list out the boat's bids also.

select s.sid, s.sname, r.bid from sailors s join reserves r on s.sid = r.sid;

20. Assume that we have a table called customer.

CustID	Name	ReferredBy
1	Neeta Sayam	
2	Dolly Dilly	1
3	Meena Kimi	2

21. Find the names of all customers who are referred by others.

22. Find the names of all customers who have referred others.

23. Find the last three customer records inserted. (Refer the above Customer table)

select * from customer order by cust_id limit 0, 3;

24. Given a table 'customer'.

CustID	Name	Age
1	Neeta Sayam	32
2	Dolly Dilly	23
3	Meena Kimi	43

How will you get rows between the range x and y where x and y will be entered by the user?

select * from customer where cust_id between 2 and 5;

25. Given three tables- sailors, boats and reserves. Sid, Bid and (Sid, Bid) are the primary keys of sailors, boats and reserves respectively. Sid and Bid are also the foreign keys of reserves which references Sid and Bid of sailors and boats relation respectively. No two sailors have same rating. The sname and bname of the sailors and boats table are cannot be null.

Sid	Sname	Rating	age
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	9	55
32	Andy	8	25.0
58	Rusty	10	35

sailors

Bid	Bname	Color
101	Interlake	Blue
102	Interlake	Red
103	Clipper	Green
104	Marine	Red

boats

Sid	Bid	Day
22	101	1/1/2004
22	102	1/1/2004
22	103	1/2/2004
31	103	5/5/2005
32	104	7/4/2005

reserves

Queries:

a) Alter the Sailor table such that age is between 18 and 40.

   update sailors set age = 30 where age not between 18 and 40;

b) Alter the Boats table such that color is Red, Blue or Green.

   update boats set color="blue" where color not in("blue", "green", "red");

c) Assuming that all the tables are created as in 1, 2 and 3 alter the table (s) such that if a record from sailors table gets deleted, then the records corresponding to the same sailor also get deleted from reserves.

   SAILORS TABLE:
   create table sailors(sid int not null primary key,
              sname varchar(50) not null,
              rating int not null,
              age int not null,
              unique(rating));

   BOATS TABLE:
   create table boats(bid int not null primary key,
              bname varchar(50) not null,
              color varchar(50) not null);

   RESERVES TABLE:

   create table reserves(sid int not null,
               bid int not null,
               day date not null,
               primary key(sid, bid),
               foreign key(sid) references sailors(sid),
               foreign key(bid) references boats(bid)
              on delete cascade
              on update cascade);

d) Drop primary key constraint from the reserves table.

ALTER TABLE `jdbc_assignmentQuestion`.`reserves`

DROP FOREIGN KEY `reserves_ibfk_1`,

DROP FOREIGN KEY `reserves_ibfk_2`;

```
ALTER TABLE `jdbc_assignmentQuestion`.`reserves`

DROP INDEX `bid`,

DROP PRIMARY KEY;
```