mysql> create table sailors(sid int not null primary key,sname varchar(50),ratin

g int not null,age float not null);

Query OK, 0 rows affected (1.75 sec)

mysql> insert into sailors(sid,sname,rating,age) values(22,"Dustin",7,45.0);

Query OK, 1 row affected (0.14 sec)

mysql> insert into sailors(sid,sname,rating,age) values(29,"Brutus",1,33.0);

Query OK, 1 row affected (0.09 sec)

mysql> insert into sailors(sid,sname,rating,age) values(31,"Lubber",9,55);

Query OK, 1 row affected (0.20 sec)

mysql> insert into sailors(sid,sname,rating,age) values(32,"Andy",8,25.0);

Query OK, 1 row affected (0.19 sec)

mysql> insert into sailors(sid,sname,rating,age) values(58,"Rusty",10,35);

Query OK, 1 row affected (0.10 sec)

mysql> select \*from sailors;

+-----+--------+--------+-----+

| sid | sname | rating | age |

+-----+--------+--------+-----+

| 22 | Dustin | 7 | 45 |

| 29 | Brutus | 1 | 33 |

| 31 | Lubber | 9 | 55 |

| 32 | Andy | 8 | 25 |

| 58 | Rusty | 10 | 35 |

+-----+--------+--------+-----+

5 rows in set (0.00 sec)

mysql> create table boats(bid int not null primary key,bname varchar(50),color v

archar(50));

Query OK, 0 rows affected (0.86 sec)

mysql> insert into boats(bid,bname,color)values(101,"Interlake","Blue");

Query OK, 1 row affected (0.16 sec)

mysql> insert into boats(bid,bname,color)values(102,"Interlake","Red");

Query OK, 1 row affected (0.45 sec)

mysql> insert into boats(bid,bname,color)values(103,"Clipper","Green");

Query OK, 1 row affected (0.55 sec)

mysql> insert into boats(bid,bname,color)values(104,"Marine","Red");

Query OK, 1 row affected (0.23 sec)

mysql> select \*from boats;

+-----+-----------+-------+

| bid | bname | color |

+-----+-----------+-------+

| 101 | Interlake | Blue |

| 102 | Interlake | Red |

| 103 | Clipper | Green |

| 104 | Marine | Red |

+-----+-----------+-------+

4 rows in set (0.06 sec)

mysql> 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);

ERROR 2006 (HY000): MySQL server has gone away

No connection. Trying to reconnect...

Connection id: 10

Current database: hcl\_batch5

Query OK, 0 rows affected (2.47 sec)

mysql> insert into reserves(sid,bid,day) values(22,101,"2004-01-01");

Query OK, 1 row affected (0.13 sec)

mysql> insert into reserves(sid,bid,day) values(22,102,"2004-01-01");

Query OK, 1 row affected (0.52 sec)

mysql> insert into reserves(sid,bid,day) values(22,103,"2004-02-01");

Query OK, 1 row affected (0.11 sec)

mysql> insert into reserves(sid,bid,day) values(31,103,"2005-05-05");

Query OK, 1 row affected (0.13 sec)

mysql> insert into reserves(sid,bid,day) values(32,104,"2005-04-07");

Query OK, 1 row affected (0.28 sec)

mysql> select \*from reserves;

+-----+-----+------------+

| sid | bid | day |

+-----+-----+------------+

| 22 | 101 | 2004-01-01 |

| 22 | 102 | 2004-01-01 |

| 22 | 103 | 2004-02-01 |

| 31 | 103 | 2005-05-05 |

| 32 | 104 | 2005-04-07 |

+-----+-----+------------+

5 rows in set (0.00 sec)

mysql> select \*from sailors;

+-----+--------+--------+-----+

| sid | sname | rating | age |

+-----+--------+--------+-----+

| 22 | Dustin | 7 | 45 |

| 29 | Brutus | 1 | 33 |

| 31 | Lubber | 9 | 55 |

| 32 | Andy | 8 | 25 |

| 58 | Rusty | 10 | 35 |

+-----+--------+--------+-----+

5 rows in set (0.00 sec)

mysql> select \*from boats;

+-----+-----------+-------+

| bid | bname | color |

+-----+-----------+-------+

| 101 | Interlake | Blue |

| 102 | Interlake | Red |

| 103 | Clipper | Green |

| 104 | Marine | Red |

+-----+-----------+-------+

4 rows in set (0.00 sec)

1🡪

mysql> select s.sname from sailors s,boats b,reserves r where s.sid=r.sid and r.

bid=b.bid and b.color="red";

+--------+

| sname |

+--------+

| Dustin |

| Andy |

+--------+

2 rows in set (0.00 sec)

2\*/

mysql> select distinct(sname) from sailors s inner join reserves r on s.sid = r.

sid;

+--------+

| sname |

+--------+

| Dustin |

| Lubber |

| Andy |

+--------+

3 rows in set (0.07 sec)

3\*/

mysql> update sailors set rating = rating\*2 where sid =(

-> select distinct(r1.sid) from reserves r1, reserves r2 where r1.day = r2.d

ay and r1.bid <> r2.bid);

Query OK, 1 row affected (0.20 sec)

Rows matched: 1 Changed: 1 Warnings: 0

4\*/

mysql> select age from sailors where sname like 'b\_%b'

-> ;

Empty set (0.09 sec)

5\*/

mysql> select sname from sailors where sid in (

-> select sid from reserves where bid in (

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

+--------+

| sname |

+--------+

| Dustin |

| Lubber |

| Andy |

+--------+

3 rows in set (0.02 sec)

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

mysql> 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 whe

re color="green" or color ="blue")));

+-----+

| sid |

+-----+

| 22 |

| 32 |

+-----+

2 rows in set (0.00 sec)

7. Find the sailors with the highest rating

mysql> select sname from sailors where rating = (select max(rating) from sailors

);

+--------+

| sname |

+--------+

| Dustin |

+--------+

1 row in set (0.09 sec)

8. Find the name of the oldest sailor.

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

+--------+

| sname |

+--------+

| Lubber |

+--------+

1 row in set (0.07 sec)

9. Count the number of different sailor names.

mysql> select count(sname) from sailors group by sname;

+--------------+

| count(sname) |

+--------------+

| 1 |

| 1 |

| 1 |

| 1 |

| 1 |

+--------------+

5 rows in set (0.06 sec)

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

mysql> select count(\*) from sailors s

-> where age>18

-> group by rating;

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

two such

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

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

+-----+

| sid |

+-----+

| 31 |

| 32 |

+-----+

2 rows in set (0.08 sec)

13. Find sailors who have not reserved any boats.

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

+-----+

| sid |

+-----+

| 29 |

| 58 |

+-----+

2 rows in set (0.00 sec)

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

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

Empty set (0.00 sec)

15. Find all the sailors older than Dustin.

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

+--------+

| sname |

+--------+

| lubber |

+--------+

1 row in set (0.02 sec)

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

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

+-------+

| sname |

+-------+

| rusty |

+-------+

1 row in set (0.00 sec)

17. Find the sailors with 3rd highest ratings.

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

+-------+

| sname |

+-------+

| andy |

+-------+

1 row in set (0.00 sec)

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

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

+-----+

| sid |

+-----+

| 22 |

| 31 |

+-----+

2 rows in set (0.00 sec)

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

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

+-----+--------+-----+

| sid | sname | bid |

+-----+--------+-----+

| 22 | dustin | 101 |

| 22 | dustin | 102 |

| 22 | dustin | 103 |

| 31 | lubber | 103 |

| 32 | andy | 104 |

+-----+--------+-----+

5 rows in set (0.00 sec)

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.

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;

1. 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;

1. 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.



Queries:

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

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

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

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

1. 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);

1. Drop primary key constraint from the reserves table.

ALTER TABLE `jdbc\_assignment`.`reserves`

DROP FOREIGN KEY `reserves\_ibfk\_1`,

DROP FOREIGN KEY `reserves\_ibfk\_2`;

ALTER TABLE `jdbc\_assignment`.`reserves`

DROP INDEX `bid` ,

DROP PRIMARY KEY;