

MYSQL TEST

MONISHA R - 80312

1. THE SUBQUERY - OPTION 4
2. MYSQLDUMP - OPTION 3
3. COLUMN LEVEL - OPTION 3
4. --HELP OPTION 3
5. PATH - OPTION 3
6. 1 CREATE PROCEDURE P () 2 BEGIN () 3 END; - OPTION B
7. ENUM - OPTION 1
8. CREATE TABLE IF NOT EXISTS employee (employeeID char(10), firstName varchar(50), lastName varchar(50), phone varchar(20), address varchar(50), PRIMARY KEY (employeeID)); - OPTION D
9. IN - OPTION 2
10. SELECT column FROM tbl - OPTION 2
11. ANSI - OPTION 2
12. DESCRIBE table; - OPTION 4
13. SELECT * FROM inventory; - OPTION 4
14. configuration files - OPTION 2
15. UNION - OPTION 3

- 16. the subquery must return a single value. - OPTION 3
- 17. show grants (displays the privileges and roles that are assigned to a MySQL user account or role) - OPTION 1
- 18. insert into cars (make, model, year) values ('Ford', 'Mustang', 2002), ('Mercedes', 'C', 2003) - OPTION 4
- 19. DROP TEMPORARY TABLE customers; - OPTION 4
- 20. foreign key; - OPTION 3

Query Writing

1. You are working with the table in this diagram. You want to use full-text search to find the customers who live on a street or a drive. What is the command to do that?

ANS - SELECT *FROM customers WHERE MATCH(address) AGAINST ('street, drive');

2. What is the valid way to create a database view in MySQL?

ANS - CREATE VIEW v1 AS SELECT * FROM t1;

3. You are working with the tables shown below. You need to generate the list of all cars, whether or not they had been sold. Which statement accomplishes that?

ANS - SELECT cars.*, purchases.date FROM cars LEFT JOIN purchases ON cars.ID = purchases.carID;

4. What steps do you need to take to normalize the table from this diagram?

ANS - Create another table to serve as a lookup for powers with fields for code and description, as well as a junction table with superhero names and power codes.

QUESTIONS

Consider the Sailors-Boats-Reserves DB described in below. S(sid, sname, rating, age) B (bid, bname, color) R (sid, bid, date)

Give a SQL expression for each of the following queries:

1. Find the colors of boats reserved by Dustin.

ANS - SELECT color FROM S, B, R WHERE R.sid=S.sid AND R.bid=B.bid AND sname='Dustin' ;

2. Find all IDs of sailors who have a rating of at least 8 or have reserved boat 103.

ANS - (SELECT sid FROM S WHERE rating>=8) UNION(SELECT sid FROM R WHERE bid=103) ;

3. Find the names of sailors who have not reserved a red boat.

ANS - SELECT S.sname FROM Sailors S, Boats B, Reserves R
WHERE B.color='red' AND B.bid=R.bid AND S.sid = R.sid;

4. Find the IDs of sailors with age over 20 who have not reserved a red boat.

ANS - SELECT S.sname FROM Sailors S, Reserves R, Boats B
WHERE B.color != 'red' and B.bid = R.bid and S.sid = R.sid and
S.sid > 20;

5. Find the names of sailors who have reserved at least two boats.

ANS - SELECT S.sname FROM Sailors S, Reserves R, Sailors S2,
Reserves R2 WHERE S.sid = R.sid and S2.sid = R2.sid and
S.sid = S2.sid and R.bid != R2.bid;

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

ANS- SELECT sname FROM s WHERE NOT EXISTS (SELECT * FROM B
WHERE NOT EXISTS (SELECT * FROM R WHERE R.sid=S.sid AND
R.bid=B.bid))

7. Find the names of sailors who have reserved all boats called Interlake.

ANS - SELECT sname FROM S WHERE (SELECT * FROM B WHERE
bname ="Interlake");

8. Find the IDs of sailors whose rating is better than some sailor called Andy.

ANS - SELECT S.sid FROM Sailors S WHERE S.rating > ANY (SELECT
S2.rating FROM Sailors S2 WHERE S2.sname = 'Andy');

9. Find the IDs of sailors whose rating is better than every sailor called Andy.

ANS - SELECT sid FROM S WHERE rating > all (SELECT rating FROM S
S2 WHERE S2.sname='Bob');

10. Find the IDs of sailors with the highest rating.

ANS - SELECT S.sid FROM Sailors S WHERE S.rating>=ALL(SELECT
S2.rating FROM Sailors S2);

11. Find the name and age of the oldest sailor

ANS - SELECT s1.sname, s1.age FROM S s1 WHERE NOT EXISTS (SELECT
*FROM S s2WHERE s2.age>s1.age) ;