**2020春《数据库系统》作业2**

主讲教师: 邹兆年(znzou@hit.edu.cn)

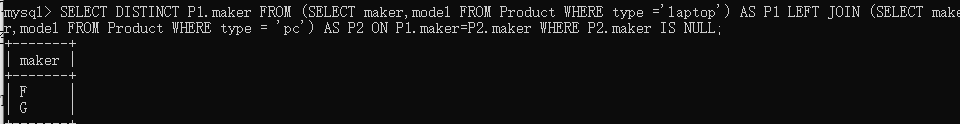
姓名: \_\_\_\_\_陶飞宇\_\_\_\_ 学号: \_\_\_\_117030720\_\_\_\_\_\_\_\_ 班号: \_\_\_\_1737301\_\_\_\_\_

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 题目 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 总分 |
| 得分 |  |  |  |  |  |  |  |  |  |  |  |

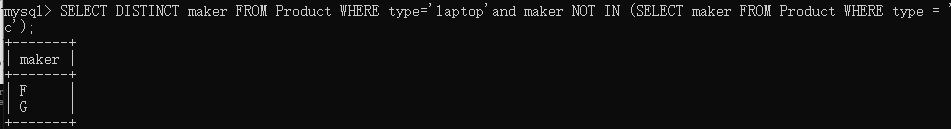
* 1. 错误，应使用DROP TABLE语句删除关系，DELETE用于删除关系的属性。
  2. 错误，前者将属性声明为主键，而后者只是使得属性不重复且非空，二者有本质上的区别。
  3. 错误，结果对A升序（默认），对B降序。
  4. 错误，SELECT不去重，投影操作去重。
  5. 错误，A中的空值也可能被计数，结果可能为|R|+1。

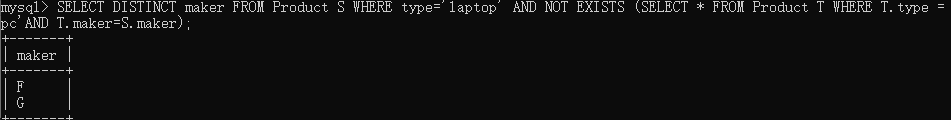
2.

a) SELECT DISTINCT P1.maker FROM (SELECT maker,model FROM Product WHERE type ='laptop') AS P1 LEFT JOIN (SELECT maker,model FROM Product WHERE type = 'pc') AS P2 ON P1.maker=P2.maker WHERE P2.maker IS NULL;

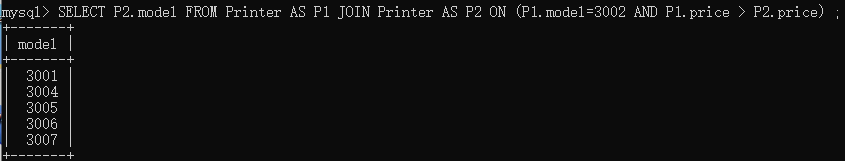


b) SELECT DISTINCT maker FROM Product WHERE type='laptop'and maker NOT IN (SELECT maker FROM Product WHERE type = 'pc');

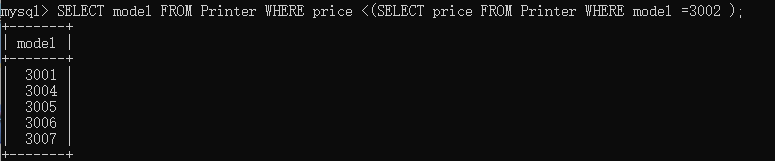


c) SELECT DISTINCT maker FROM Product S WHERE type='laptop' AND NOT EXISTS (SELECT \* FROM Product T WHERE T.type = 'pc'AND T.maker=S.maker); 

d) SELECT P2.model FROM Printer AS P1 JOIN Printer AS P2 ON (P1.model=3002 AND P1.price > P2.price) ;



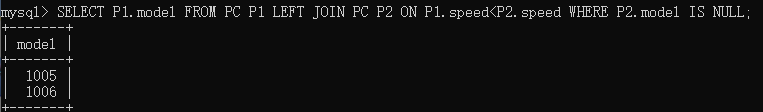
e) SELECT model FROM Printer WHERE price <(SELECT price FROM Printer WHERE model =3002 );



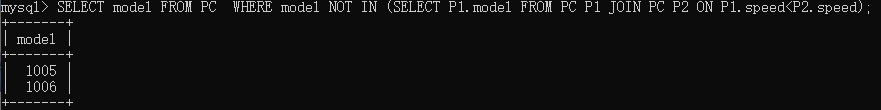
f) SELECT model FROM Printer P1 WHERE EXISTS (SELECT \* FROM Printer P2 WHERE P2.model =3002 AND P2.price >P1.price );



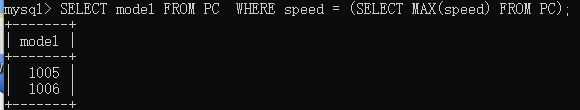
g) SELECT P1.model FROM PC P1 LEFT JOIN PC P2 ON P1.speed<P2.speed WHERE P2.model IS NULL;



h) SELECT model FROM PC WHERE model NOT IN (SELECT P1.model FROM PC P1 JOIN PC P2 ON P1.speed<P2.speed);



i) SELECT model FROM PC WHERE speed = (SELECT MAX(speed) FROM PC);



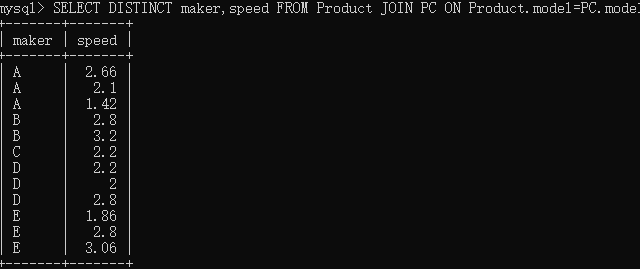
j) SELECT model FROM PC WHERE speed >= ALL(SELECT speed FROM PC);



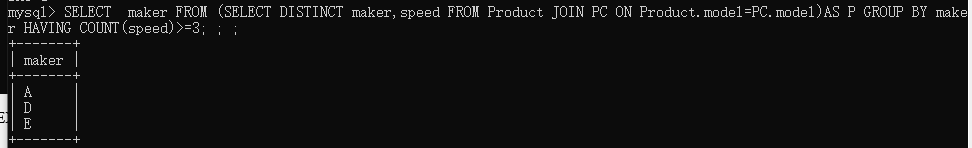
k) SELECT model FROM PC P1 WHERE NOT EXISTS (SELECT \* FROM PC P2 WHERE P1.speed<P2.speed);



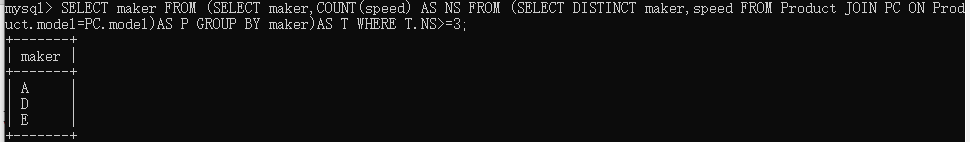
l) SELECT DISTINCT maker,speed FROM Product JOIN PC ON Product.model=PC.model ;



m) SELECT maker FROM (SELECT DISTINCT maker,speed FROM Product JOIN PC ON Product.model=PC.model)AS P GROUP BY maker HAVING COUNT(speed)>=3;



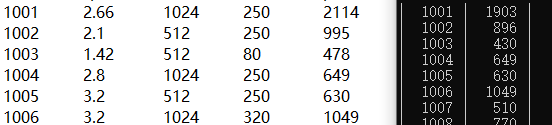
n) SELECT maker FROM (SELECT maker,COUNT(speed) AS NS FROM (SELECT DISTINCT maker,speed FROM Product JOIN PC ON Product.model=PC.model)AS P GROUP BY maker)AS T WHERE T.NS>=3;



o) UPDATE PC,Product SET PC.price=0.9\*PC.price WHERE Product.model=PC.model AND Product.maker='A';

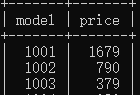
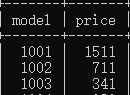
→

p) UPDATE PC SET price = 0.9\*price WHERE model IN (SELECT a.model FROM (SELECT PC.model FROM Product JOIN PC ON (Product.model=PC.model AND Product.maker='A'))AS a);



(1001\1002\1003都减少了10%)

q) UPDATE PC SET PC.price=0.9\*PC.price WHERE EXISTS (SELECT \* FROM Product WHERE Product.model=PC.model AND Product.maker='A');

→

3.

g) SELECT P1.model FROM PC P1 LEFT JOIN PC P2 ON P1.speed<P2.speed WHERE P2.model IS NULL;

h) SELECT model FROM PC WHERE model NOT IN (SELECT P1.model FROM PC P1 JOIN PC P2 ON P1.speed<P2.speed);

i) SELECT model FROM PC WHERE speed = (SELECT MAX(speed) FROM PC);

j) SELECT model FROM PC WHERE speed >= ALL(SELECT speed FROM PC);

k) SELECT model FROM PC P1 WHERE NOT EXISTS (SELECT \* FROM PC P2 WHERE P1.speed<P2.speed);

g的方法为外连接，只需要连接出一张表之后找出其中P2.speed为空的项目，复杂度较低，但是可读性欠佳

h的方法为NOT IN，该方法需要将每一个元组以此与一张表的每一个元组进行比较，复杂度大，当数据基数过大时可能导致崩溃，但是可读性好，逻辑清晰

i为直接的等值比较，但是MAX函数需要排序，根据数据的顺序混乱程度决定其复杂度，该方法的可读性也是较好的。

g与h相似，需要一个个比较，复杂度为N^2,但是可读性好，直观易懂。

k与g、h相似，同样需要遍历，使用了双重否定，可读性也欠佳。