**Teste de Sql para vaga de analista de BI Jr. TradeMaster**

1) Seu Joaquim está com problemas em sua venda, onde sem querer foi cadastrado em duplicidades alguns produtos na tabela de produtos. Escreva uma query que mostre a quantidade de vezes que o produto está em duplicidade e o nome do produto.

|  |  |
| --- | --- |
| Coluna | Tipo |
| id | Int |
| produto | String |
| prateleira | String |
| preco | Float |

R)

**SELECT produto, COUNT(\*)**

**FROM tabela\_produtos  
GROUP BY produto  
HAVING COUNT(\*) > 1**

2) Uma tabela de funções, contém duas colunas: X e Y.

|  |  |
| --- | --- |
| Coluna | Tipo |
| X | Int |
| Y | Int |

Assim são 2 pares (X1 e Y1) e (X2 e Y2) São considerados pares simétricos se X1 = Y2 e X2 = Y1.

Escreva uma query que retorne todos os pares simétricos em ordem decrescente pelo valor de X. Exiba as linhas de forma que X1 <= Y1

R)

**SELECT X, Y FROM (**

**SELECT X, Y FROM Functions WHERE X=Y GROUP BY X, Y HAVING COUNT(\*)=2**

**UNION**

**SELECT f1.X, f1.Y FROM Functions f1, Functions f2**

**WHERE f1.X < f1.Y**

**AND f1.X=f2.Y**

**AND f2.X=f1.Y**

**)t**

**ORDER BY X, Y;**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

R)

**SELECT X, Y FROM (**

**SELECT X, Y FROM Functions WHERE X=Y GROUP BY X, Y HAVING COUNT(\*)=2**

**UNION**

**SELECT f1.X, f1.Y FROM Functions f1**

**WHERE f1.X < f1.Y**

**AND EXISTS (SELECT f2.X, f2.Y FROM Functions f2 WHERE f1.X=f2.Y AND f1.Y=f2.X)**

**)t**

**ORDER BY X, Y;**

----------------------

3) Você é um Cavalheiro, personagem principal de um jogo de rpg, e está buscando uma espada mais poderosa para atacar seus adversários. Obviamente você quer a melhor espada, pagando o mínimo em moedas e que esteja em boas condições. Portanto escreva uma query que exiba o id, a idade da espada, as moedas necessárias e a força, mostrando a espada mais poderosa primeiro que não seja almadiçoada.

Na primeira tabela (espadas) você possui os seguintes dados:

id (numero único da espada)

codigo (usado como chave estrangeira para outra tabela)

moedas (preço da espada)

força (o quanto a espada é forte / dano da espada)

|  |  |
| --- | --- |
| Coluna | Tipo |
| id | Int |
| codigo | Int |
| moedas | Int |
| Forca | Int |

Segunda tabela (dados\_espadas)

codigo (usado como chave estrangeira para outra tabela)

idade (relativo a quão velho a arma é)

almadiçoada (se ao tocar a arma você ficará almadiçoado; isso não é bom! Onde 0 significa não almadiçoada e 1 significa almadiçoada)

tabela abaixo:

|  |  |
| --- | --- |
| Coluna | Tipo |
| codigo | Int |
| idade | Int |
| almadicoada | Int |

R)

**SELECT espadas.id, dados\_espadas.idade, espadas.moedas, espadas.força**

**FROM espadas**

**INNER JOIN dados\_espadas**

**ON espadas.codigo = dados\_espadas\_codigo**

**WHERE dados\_espadas.almadiçoada = 0**

**ORDER BY espadas.força DESC**

------------------------------------------------