

1) Criar uma função em PLpgSQL que mediante os parâmetros enviados (CNH e Velocidade Apurada) se possa testar se um motorista deverá receber ou não Multa. Caso tenha multa o resultado é a inserção de um novo registro na tabela 'ex_multa'.

- O procedimento deve receber dois valores por parâmetro 'CNH' do motorista e a 'velocidadeApurada';
- O cálculo da pontuação do motorista é efetuado da seguinte forma:
- Se a velocidade estiver entre 80.01 e 110, o motorista deve ser multado em R\$ 120,00 e receber 20 pontos;
- Se a velocidade estiver entre 110.01 e 140, o motorista deve ser multado em R\$ 350,00 e receber 40 pontos;
- Se a velocidade estiver acima de 140, o motorista deve ser multado em R\$ 680,00 e receber 60 pontos;
- O sistema deve considerar somente 90% da velocidade apurada para o cálculo da multa.

2) Escreva outro procedimento que atualize o campo 'totalMultas', em reais, da tabela 'ex_motorista' a partir dos totais apurados para cada motorista autuado na tabela 'ex_multa'.

- Motorista sem multa deverão possuir o valor 0.00 em 'totalMultas';
- Não duplicar valores caso a rotina seja disparada novamente.

3) Procedimento para gerar um totalizador por categoria de multas, bem como mês/ano de todas as multas ocorridas, considerando a quantidade total de pontos, o valor total de multas apuradas.

- 20 pontos = categoria 1;
- 40 pontos = categoria 2;
- 60 pontos = categoria 3.

Rever os procedimentos e prever uma nova tabela chamada 'ex_categoria'.

Faz parte do trabalho a melhor adequação a esta necessidade.

OBS.: A resposta das questões 1, 2 e 3 estão em função única na página seguinte:

CREATE OR REPLACE FUNCTION

```
fn_verificaMulta(varCNH char(10), velocidadeApurada int) RETURNS char(100) AS $$
```

DECLARE

```
_calculo decimal;
```

```
_nome char(20);
```

```
_msg char(100);
```

BEGIN

```
_calculo = (velocidadeApurada*0.9);
```

```
_nome = (    select ex_motorista.nome  
            from ex_motorista  
            where ex_motorista.cnh = fn_verificaMulta.varCNH    );
```

```
_msg = 'NÃO FOI MULTADO: ' || _nome || ' | CNH: ' || varCNH || ' | Velocidade Calculada: '  
|| _calculo || 'Km/h';
```

IF _calculo > 80.01 and _calculo < 110 THEN

```
    insert into ex_multa (cnh, velocidadeApurada, velocidadeCalculada, pontos, valor,  
data) values (varCNH, velocidadeApurada, _calculo, 20, 120,  
CURRENT_TIMESTAMP);  
    insert into ex_categoria (cnh, categoria) values (varCNH, 1);  
    update ex_motorista  
    set totalMultas = (totalMultas +120)  
    where ex_motorista.cnh = fn_verificaMulta. varCNH;  
    return 'MULTADO: ' || _nome || ' | CNH: ' || varCNH || ' | Pontos: 20 | Velocidade  
Calculada: ' || _calculo || 'Km/h | (base de 80Km/h a 110Km/h)';
```

ELSEIF _calculo > 110.01 and _calculo < 140 THEN

```
    insert into ex_multa (cnh, velocidadeApurada, velocidadeCalculada, pontos, valor,  
data) values (varCNH, velocidadeApurada, _calculo, 40, 350,  
CURRENT_TIMESTAMP);  
    insert into ex_categoria (cnh, categoria) values (varCNH, 2);  
    update ex_motorista  
    set totalMultas = (totalMultas +350)  
    where ex_motorista.cnh = fn_verificaMulta. varCNH;  
    return 'MULTADO: ' || _nome || ' | CNH: ' || varCNH || ' | Pontos: 40 | Velocidade  
Calculada: ' || _calculo || 'Km/h | (base de 110/h a 140/h)';
```

ELSEIF _calculo > 140 THEN

```
    insert into ex_multa (cnh, velocidadeApurada, velocidadeCalculada, pontos, valor,  
data) values (varCNH, velocidadeApurada, _calculo, 60, 680,  
CURRENT_TIMESTAMP);  
    insert into ex_categoria (cnh, categoria) values (varCNH, 3);  
    update ex_motorista  
    set totalMultas = (totalMultas +680)  
    where ex_motorista.cnh = fn_verificaMulta. varCNH;  
    return 'MULTADO: ' || _nome || ' | CNH: ' || varCNH || ' | Pontos: 60 | Velocidade  
Calculada: ' || _calculo || 'Km/h | (base > 140Km/h)';
```

```
END IF;
```

```
RETURN _msg;
```

```
END;
```

```
$$ LANGUAGE PLpgSQL;
```

- 4) Escreva outro procedimento que liste o totalizador por categoria, considerando os valores pontuados e valores multados em reais.

```
select categoria, count(categoria) as totalCategoria, sum(valor) as totalValor,  
sum(pontos) as totalPontos  
from ex_categoria, ex_multa  
where ex_categoria.cnh = ex_multa.cnh  
and ex_categoria.id = ex_multa.id  
group by categoria  
order by categoria;
```