

Processo Seletivo GRIS 2020

Nome: Leonardo Andrade

TAG 2 Banco de Dados - Análise de um Banco de dados

1) Qual foi o total de compras feitas nesse mercado?

R: Temos a tabela “compra” no nosso banco de dados que lista todas as compras feitas nesse mercado, vamos utilizar o comando **SELECT DISTINCT COUNT(*) FROM compra;** e teremos então como retorno o total de compras efetuadas, que é 50000.

2) Qual é o nome do primeiro comprador, em ordem alfabética, que tenha nascido após 1990?

R: O que pensei primeiramente foi em utilizar o comando **SELECT * FROM fregueses WHERE data_nasc >= '1991' ORDER BY data_nasc,nome DESC;** para obter a lista de fregueses que nasceram em 1991 em diante, ordenando pela data de nascimento e nome. Com isso, obtive que a primeira freguesa que nasceu após o ano de 1990 atende aos seguintes dados: *idfregueses: '371', nome: 'samanta', contato: '921480118', data_nasc:'1991-02-27'*. Porém ainda em dúvida, decide explorar um pouco mais usando o comando:

```
SELECT DISTINCT fregueses.idfregueses, fregueses.nome,fregueses.data_nasc, compra.produto
```

```
FROM fregueses JOIN compra ON compra.comprador = fregueses.idfregueses WHERE data_nasc >= '1991'
```

```
ORDER BY nome,data_nasc;
```

Com isso minha intenção era verificar se de fato todos os fregueses haviam efetuado compras, fazendo uso da junção das tabelas *fregueses* e *compra* com a chave estrangeira *comprador*. Implementei também a busca em ordem alfabética para ter o primeiro nessa ordem a nascer após 1990. Assim, quem obtive foi *idfregueses:'349', nome:'abel', data_nasc:'1996-04-12', produto:'3838'*.

3)Qual é o nome do comprador mais jovem?

R: Para resolver essa questão fiz uso de um comando semelhante ao usado na questão anterior

```
SELECT DISTINCT fregueses.nome, fregueses.data_nasc, fregueses.idfregueses, compra.produto
```

```
FROM fregueses JOIN compra ON compra.comprador = fregueses.idfregueses ORDER BY data_nasc DESC;
```

Esse comando se difere do anterior no quesito de que ele junta as tabelas, porém apenas ordena por data de nascimento por ordem decrescente. Com isso obtive o seguinte comprador mais jovem: *nome:'iranice', data_nasc:'2010-12-10', idfregueses:'56', produto: '16153'*.

4)Qual o produto tem melhor avaliação e menor preço?

R: Utilizando o comando

```
SELECT DISTINCT produto.idproduto, produto.preco, `avaliacao produto`.nota
FROM produto JOIN `avaliacao produto`
ON `avaliacao produto`.produto = produto.idproduto
WHERE NOTA = '10'
ORDER BY preco;
```

Minha intenção foi juntar as tabelas *produto* e *avaliacao produto* usando a chave estrangeira *produto*, com o fim de listar todos os produtos mais bem avaliados (nota = 10) e ordená-los por preço do menor para o maior. Dessa forma, obtive o seguinte resultado: *idproduto: '15691', preco: '10.00', nota: '10'*.

5)Qual é o vendedor que possui melhor avaliação média e maior quantidade de vendas?

R: Não consegui chegar a uma resposta concreta nessa questão. Tinha em mente juntar dois *SELECTS*:

```
SELECT produto.vendedor, vendedor.nome, COUNT(compra.produto)
FROM vendedor JOIN produto ON vendedor.idvendedor = produto.vendedor
JOIN compra ON produto.idproduto = compra.produto
GROUP BY vendedor.idvendedor
ORDER BY COUNT(produto.idproduto) DESC;
```

O comando acima me retorna a lista de vendedores com seus IDs e a quantidade de produtos que foi vendido por CADA UM deles, tudo ordenado pelo que mais vendeu para que menos vendeu. Usei o COUNT() para contar a quantidade de produtos e agrupei pelo ID dos vendedores. Com isso, verifiquei que a campeã de vendas é a *vendedor: '931', nome: 'maysa', produtos vendidos por ela: '98'*. Basicamente o que fiz foi juntar as tabelas *vendedor*, *produto* e *compra*, poderia juntar apenas *vendedor* e *produto* e analisar qual vendedor tem mais produtos ligados a si, porém isso não significa necessariamente que ele vendeu toda a mercadoria referente ao seu estoque. Logo, para termos certeza, melhor foi analisar a partir da tabela *compra* também.

Para verificação da maior média de avaliação, utilizei:

```

SELECT vendedor.idvendedor, vendedor.nome, AVG(`avaliacao vendedor`.nota)
FROM vendedor JOIN `avaliacao vendedor` ON vendedor.idvendedor = `avaliacao vendedor`.vendedor
GROUP BY vendedor.idvendedor
ORDER BY AVG(`avaliacao vendedor`.nota) DESC;

```

Nesse caso fiz um SELECT juntando as tabelas *vendedor* e *avaliacao vendedor*. Usei o comando AVG() para calcular a média e agrupei os vendedores pelo ID, ordenando decrescentemente. Dessa forma, verifiquei que a vendedora mais bem avaliada atende pelas seguintes credenciais *idvendedor: '33', nome: 'carol\n', nota média: '6.837837837837838'*.

Numa tentativa de juntar os dois e obter um resultado só do mais bem avaliado e com mais produtos vendidos, utilizei o comando:

```

SELECT      produto.vendedor,      vendedor.nome,      COUNT(compra.produto),
AVG(`avaliacao vendedor`.nota)
FROM vendedor JOIN produto ON vendedor.idvendedor = produto.vendedor
JOIN compra ON produto.idproduto = compra.produto
JOIN `avaliacao vendedor` ON vendedor.idvendedor = `avaliacao vendedor`.vendedor
GROUP BY vendedor.idvendedor
ORDER BY COUNT(produto.idproduto) DESC, AVG(`avaliacao vendedor`.nota)
DESC;

```

Tentei juntar as tabelas *vendedor*, *produto* e *compra* a fim de listar por nota de avaliação média e quantidade de produtos vendidos, ambos em ordem decrescente. O que apareceu no topo da listagem foi *idvendedor: '931', nome: 'maysa', produtos vendidos: '9604', média de avaliação: '5.795918367346939'*. Quando olhei a quantidade de produtos vendidos percebi que essa consulta não havia sido bem sucedida. Pensei em algumas soluções com UNION e AND, mas todas fracassaram.

