

# Teste de SQL

Considere a seguinte tabela:

## Tabela de produtos

Campo	Tipo de Campo	Chave
cod_prod	Integer (8)	X
loj_prod	Integer (8)	X
desc_prod	Char (40)	
dt_inclu_prod	Data (dd/mm/yyyy)	
preco_prod	decimal (8,3)	

Com base na tabela de “produtos” acima favor inserir um registro na referida tabela passando os seguintes valores : cod\_prod =170, loj\_prod=2, desc\_prod=LEITE CONDESADO MOCOCA, dt\_inclu\_prod=30/12/2010 e preço\_prod = R\$45,40.

```
INSERT INTO produtos (cod_prod, loj_prod, dt_inclu_prod, preco_prod) VALUES (170, 2,'LEITE CONDENSADO MOCOCA', '2010-12-30',95.40)
```

O Índice da tabela de “produtos” é o cód\_prod e a loj\_prod, com base no referido índice faça a alteração do preço do produto para R\$95,40, lembrando que o cod\_prod =170 e a loj\_prod=2:

```
UPDATE produtos SET = 95.40 preco_prod WHERE cod_prod = 170 AND loj_prod = 2;
```

Com base na tabela de “produtos” monte um select trazendo todos os registros da loja 1 e 2:

```
SELECT * FROM produtos WHERE loj_prod IN (1,2);
```

Com base na tabela de “produtos” monte um select para trazer a maior e a menor data de inclusão do produto “dt\_inclu\_prod”:

```
SELECT MAX(dt_inclu-peod) AS maior_data, MIN(dt_inclu_prod) AS menor_data from produtos;
```

Com base na tabela de “produtos” monte um select para trazer a quantidade total de registros existentes na tabela de “produtos”:

```
SELECT COUNT (*) AS total_registros FROM produtos;
```

Com base na tabela de “produtos” monte um select para trazer todos os produtos que comecem com a letra “L” na tabela de “produtos”:

```
Select * from produtos Where desc_prod LIKE 'L%';
```

Com base na tabela de “produtos” monte um select para trazer a soma de todos os preços dos produtos totalizado por loja:

```
Select loj_prod, sum(preco_prod) as total_precos from produtos group by loj_prod;
```

Com base na tabela de “produtos” monte um select para trazer a soma de todos os preços dos produtos totalizados por loja que seja maior que R\$100.000

```
Select loj_prod, SUM(preco_prod) as total_precos from produtos group by loj_prod having sum(preco_prod) > 100000;
```

Observe as Tabelas Abaixo:

### Tabela de Produtos

Campo	Tipo de Campo	Chave	Comentário
Cód_prod	Integer (8)	X	Código do Produto
loj_prod	Integer (8)	X	Código da Loja
desc_prod	Char (40)		Descrição do Produto
Dt_inclu_prod	Data (dd/mm/yyyy)		Data de Inclusão do Produto
preco_prod	decimal (8,3)		Preço do Produto

### Tabela de Estoque

Campo	Tipo de Campo	Chave	Comentário
Cód_prod	Integer (8)	X	Código do Produto
loj_prod	Integer (8)	X	Código da Loja
qtd_prod	decimal(15,3)		Quantidade em Estoque do Produto

### Tabela de Lojas

Campo	Tipo de Campo	Chave	Comentário
loj_prod	Integer (8)	X	Código da Loja
desc_loj	Char (40)		Descrição da Loja

A) Montar um unico select para trazer os seguintes campos: o código da loja do produto, a descrição da loja, código do produto, a descrição do produto, o preço do produto, a quantidade em estoque do produto. Considere que o código da loja para esta consulta seja igual a 1.

```
SELECT p.loj_prod, l.desc_loj, p.cod_prod, p.desc_prod, p.preco_prod, e.qtd_prod FROM
produtos p INNER JOIN estoque e ON p.cod_prod = e.cod_prod AND p.loj_prod =
e.loj_prod INNER JOIN lojas l ON p.loj_prod = l.loj_prod WHERE p.loj_prod = 1;
```

B) Observe a estrutura da tabela de estoque e da tabela de produtos, monte um select para trazer todos os produtos que existem na tabela de produtos que não existem na tabela de estoque.

```
SELECT p.*
FROM produtos p
LEFT JOIN estoque e ON p.cod_prod = e.cod_prod AND p.loj_prod = e.loj_prod
WHERE e.cod_prod IS NULL;
```

C) Observe a estrutura da tabela de estoque e da tabela de produtos, monte um select para trazer todos os produtos que existem na tabela de estoque que não existem na tabela de produtos.

```
SELECT e.* FROM estoque e LEFT JOIN produtos p ON e.cod_prod AND e.loj_prod =  
p.loj_prod WHERE p.cod_prod IS NULL;
```