

# 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.

resposta:

```
INSERT INTO produtos(cod_prod, loj_prod, desc_prod, dt_inclu_prod, preco_prod)
VALUES(170, 2, "LEITE CONDESADO MOCOCA", "30/12/2010", 45,40);
```

O Índice da tabela de “produtos” é o cod\_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:

resposta:

```
UPDATE produtos
SET preco_prod=95,40
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:

resposta:

```
SELECT * FROM produtos;
```

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

resposta:

```
SELECT MAX(dt_inclu_prod) AS MaiorData
FROM produtos;
```

```
SELECT MIN(dt_inclu_prod) AS MenorData
FROM produtos;
```

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

resposta:

```
SELECT COUNT(*) 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”:

resposta:

```
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 SUM(preco_prod) AS total  
FROM produtos;
```

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

resposta:

```
SELECT SUM(preco_prod) AS total  
FROM produtos  
WHERE total > 100,000;
```

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_pro d	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 único 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.

resposta:

```
SELECT a.loj_prod, b.desc_loj, a.cod_prod, a.desc_prod, a.preco_prod, c.qtd_prod  
FROM produtos AS a, loja AS b, estoque AS c  
WHERE a.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.

resposta:

```
SELECT * FROM produtos e  
WHERE e.cod_prod  
NOT IN (SELECT cod_prod FROM estoque);
```

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.

resposta:

```
SELECT cod_prod FROM estoque e  
WHERE e.cod_prod  
NOT IN (SELECT cod_prod FROM produtos);
```