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 VALUES (170, 2, 'LEITE CONDENSADO
MOCOCA', '30-12-2010', 45.40);
```

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 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:

```
SELECT * FROM produtos WHERE loj_prod=1 OR loj_prod=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 * FROM produtos ORDER BY dt_inclu_prod DESC LIMIT 1
UNION SELECT * FROM produtos ORDER BY dt_inclu_prod LIMIT 1;
```

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 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 WHERERER 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) FROM produtos WHERE loj_prod=1 UNION
ALL SELECT SUM(preco_prod) FROM produtos WHERE loj_prod=2;
```

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 SUM(preco_prod) FROM produtos WHERE loj_prod=1 AND
preco_prod >100.00 UNION ALL SELECT SUM(preco_prod) FROM
produtos WHERE loj_prod=2 AND preco_prod >100.00;
```

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 cod_prod, desc_prod, qtd_prod, loj_prod, desc_loj FROM
produtos, produtos, estoque, lojas, lojas WHERE 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 a.cod_prod, b.cod_prod FROM produtos AS A LEFT JOIN
estoque AS B ON a.cod_prod = b.cod_prod WHERE b.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 a.cod_prod, b.cod_prod FROM produtos AS A RIGHT JOIN
estoque AS B ON a.cod_prod = b.cod_prod WHERE a.cod_prod is
NULL;
```