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) |  |

**CREATE** **DATABASE** **IF** **NOT** **EXISTS** teste\_sql;

**USE** teste\_sql;

**CREATE** **TABLE** **IF** **NOT** **EXISTS** produtos(

cod\_prod **int**(8),

loj\_prod **int**(8),

desc\_prod **varchar**(40),

dt\_inclu\_prod **date**,

preco\_prod **decimal**(8,3),

**PRIMARY** **KEY** (cod\_prod, loj\_prod)

);

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, desc\_prod, dt\_inclu\_prod, preco\_prod)

**VALUES**

(170, 2, 'LEITE CONDENSADO MOCOCA', "2010/12/30", 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 **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”:

#apenas a menor e maior data

**SELECT** **MIN**(dt\_inclu\_prod) **AS** menor\_data, **MAX**(dt\_inclu\_prod) **AS** maior\_data

**FROM** produtos;

#todo o registro dessas datas

**SELECT** \*

**FROM** produtos

**WHERE** dt\_inclu\_prod = (**SELECT** **MAX**(dt\_inclu\_prod) **FROM** produtos)

**OR** dt\_inclu\_prod = (**SELECT** **MIN**(dt\_inclu\_prod) **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** quantidade\_produtos

**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** **SUM**(preco\_prod) **AS** preco\_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

**SELECT** loj\_prod, **SUM**(preco\_prod)

**AS** valor\_total\_prod

**FROM** produtos

**GROUP** **BY** loj\_prod

**HAVING** valor\_total\_prod > 100;

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 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**CREATE** **TABLE** **IF** **NOT** **EXISTS** estoques(

cod\_prod **int**(8),

loj\_prod **int**(8),

qtd\_prod **decimal**(15,3),

**PRIMARY** **KEY** (cod\_prod, loj\_prod)

);

**CREATE** **TABLE** **IF** **NOT** **EXISTS** lojas(

loj\_prod **int**(8),

desc\_loj **varchar**(40),

**PRIMARY** **KEY** (loj\_prod)

);

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 **AS** p

**JOIN** lojas **AS** l **ON** p.loj\_prod = l.loj\_prod

**JOIN** estoques **as** e **on** l.loj\_prod = e.loj\_prod **and** e.cod\_prod = p.cod\_prod

**WHERE** l.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** \* **FROM** produtos

**WHERE** (cod\_prod, loj\_prod) **NOT** **IN** (

**SELECT** cod\_prod, loj\_prod

**FROM** estoques

)

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.

#para saber a o produto q não existe em produtos com a desc\_prod de outras lojas

**SELECT** p.desc\_prod, e.\* **FROM** estoques **AS** e

**JOIN** produtos **AS** p **ON** p.cod\_prod = e.cod\_prod

**WHERE** (e.cod\_prod , e.loj\_prod) **NOT** **IN** (

**SELECT** cod\_prod, loj\_prod

**FROM** produtos

)

#para saber sem a desc\_prod

**SELECT** \* **FROM** estoques

**WHERE** (cod\_prod, loj\_prod) **NOT** **IN** (

**SELECT** cod\_prod, loj\_prod

**FROM** produtos

)