Nome: Matheus Vieira Silva
E-mail: matheusvieira.contato@hotmail.com
Tel: +55 (13) 97408-6577

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 T_PRODUTO (
170,
2,
‘LEITE CONDESADO MOCOCA’,
TO_DATE(‘2010/12/31,’YYYY/MM/DD)’,
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 T_PRODUTO SET preco_prod = 95.40 WHERE cod_prod = 170;
```

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

Obs: Só possui somente uma loja de acordo com o formulário de teste. Sendo assim, inseri mais dados na tabela para que haja uma melhor manipulação de dados de acordo com o teste.

```
SELECT * FROM T_PRODUTO 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 * FROM T_PRODUTO WHERE dt_inclu_prod = (SELECT MIN(dt_inclu_prod) FROM T_PRODUTO);

SELECT * FROM T_PRODUTO WHERE dt_inclu_prod = (SELECT MAX(dt_inclu_prod) FROM T_PRODUTO);

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

SELECT COUNT(*) FROM T_PRODUTO;

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 T_PRODUTO 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 FROM T_PRODUTO 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 * FROM T_PRODUTO WHERE preco_prod > 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_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.

Obs: Criei uma coluna na tabela estoque chamada cd_estoque, para que haja um relacionamento entre as entidades e então fazendo uma consulta e manipulação de dados mais coerente.

```
SELECT T_PRODUTO.cod_loj, T_PRODUTO.desc_loj, T_PRODUTO.cod_prod,
T_PRODUTO.desc_prod, T_PRODUTO.preco_prod, T_ESTOQUE.qtd_prod
FROM T_PRODUTO, T_ESTOQUE WHERE T_PRODUTO.cod_prod =
T_ESTOQUE.loj_prod;
```

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.cod_prod  "CODIGO PRODUTO",
      P.loj_prod  "Produto Loja",
      P.desc_prod "Descrição do produto",
      P.dt_inclu_prod "Data produto",
      P.preco_prod "Preco Produto"
FROM T_PRODUTO P, T_ESTOQUE I
WHERE P.cod_prod = I.cod_prod (+) AND I.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 F.cod_prod  "Codigo produto",
       I.cd_estoque "Codigo estoque",
       I.cod_prod "Codigo produto no estoque",
       I.loj_prod "Estoque produto loja",
       I.qtd_prod "Quantidade Estoque"
FROM T_ESTOQUE I RIGHT OUTER JOIN
     T_PRODUTO F ON ( F.cod_prod = I.cod_prod )
     WHERE I.cod_prod IS NULL;
```

GitHub: https://github.com/matheus-vieiras/avaliacao-titan-sql-java