

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)	

```
-----
-- Schema testsql
-----

CREATE SCHEMA IF NOT EXISTS `testsql` DEFAULT CHARACTER SET utf8 ;
USE `testsql` ;

-----

-- Table `testsql`.`loja`
-----

CREATE TABLE IF NOT EXISTS `testsql`.`loja` (
  `cod_loja` INT(8) NOT NULL AUTO_INCREMENT,
  `descricao_loja` CHAR(40) NOT NULL,
  PRIMARY KEY (`cod_loja`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

-----

-- Table `testsql`.`productos`
-----

CREATE TABLE IF NOT EXISTS `testsql`.`productos` (
  `cod_prod` INT(8) NOT NULL AUTO_INCREMENT,
  `descricao_producto` CHAR(40) NULL DEFAULT NULL,
  `data_inclusao` DATE NULL DEFAULT NULL,
  `preco_producto` DECIMAL(8,3) NULL DEFAULT NULL,
  `id_loja` INT(8) NOT NULL,
  PRIMARY KEY (`cod_prod`, `id_loja`),
  INDEX `fk_productos_loja_idx` (`id_loja` ASC) VISIBLE,
  CONSTRAINT `fk_productos_loja`
    FOREIGN KEY (`id_loja`)
      REFERENCES `testsql`.`loja` (`cod_loja`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
```

```

-----
-- Table `testsql`.`estoque`
-----
CREATE TABLE IF NOT EXISTS `testsql`.`estoque` (
  `qtd_prod` DECIMAL(15,3) NULL DEFAULT NULL,
  `id_prod` INT(8) NOT NULL,
  `id_loja` INT(8) NOT NULL,
  PRIMARY KEY (`id_prod`, `id_loja`),
  INDEX `fk_estoque_productos1_idx` (`id_prod` ASC, `id_loja` ASC)
  VISIBLE,
  CONSTRAINT `fk_estoque_productos1`
    FOREIGN KEY (`id_prod`, `id_loja`)
    REFERENCES `testsql`.`productos` (`cod_prod`, `id_loja`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

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 `testsql`.`productos` (`cod_prod`, `descricao_producto`,
`data_inclusao`, `preco_producto`, `id_loja`) VALUES ('170', 'LEITE
CONDESADO MOCOCA', '2010-12-30', '45,40', '2');

```

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:

```

UPDATE `testsql`.`productos` SET `preco_producto` = '95.40' WHERE
(`cod_prod` = '170') and (`id_loja` = '2');

```

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

```

USE testsql;
SELECT * FROM productos, loja
WHERE productos.id_loja=loja.cod_loja AND loja.cod_loja < 3;

```

cod_prod	descricao_producto	data_inclusao	preco_producto	id_loja	cod_loja	descricao_loja
1	boneca	2010-12-30	1000.000	1	1	brinquedos
4	cavalinho madeira	2011-11-12	1500.000	1	1	brinquedos
170	Leite condesado mococa	2010-12-30	95.400	2	2	alimentos

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

Maior data

```
SELECT cod_prod,MAX(data_inclusao)
FROM productos;
```

Menor data

```
SELECT cod_prod,MIN(data_inclusao)
FROM productos;
```

	cod_prod	MIN(data_inclusao)
1	1	2009-01-02

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

```
SELECT cod_prod,COUNT(*)
FROM productos;
```

	cod_prod	COUNT(*)
1	1	6

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 descricao_produto
FROM productos
WHERE descricao_produto LIKE 'L%';
```

descricao_produto
Leite condensado mococa

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

```
SELECT cod_prod,SUM(preco_produto)
FROM productos
GROUP BY id_loja;
```

cod_prod	SUM(preco_produto)
1	2500.000
170	95.400
2	150.000
5	250.000

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 cod_prod,SUM(preco_produto)
FROM productos
WHERE preco_produto > 100;
```

Observe as Tabelas Abaixo:

Tabela de Produtos

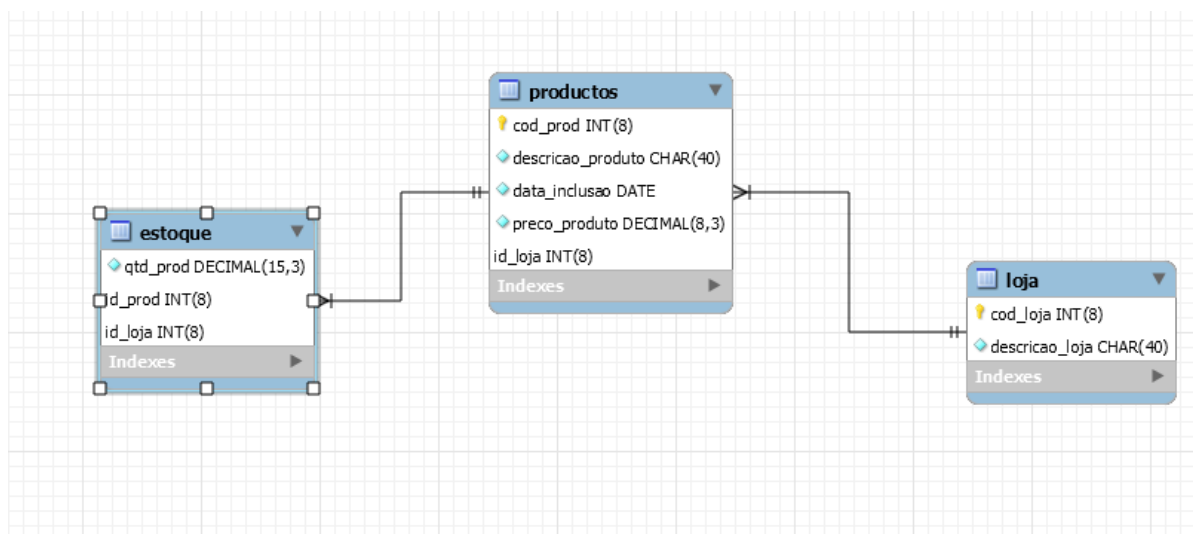
Campo	Tipo de Campo	Chave	Comentário
Cód_prod	Integer (8)	X	Código do Produto
loj_prod	Integer (8)	X	Código da Loja
desc_prod	Char (40)		Descrição do Produto
Dt_inclu_prod	Data (dd/mm/yyyy)		Data de Inclusão do Produto
preco_prod	decimal (8,3)		Preço do Produto

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



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 * FROM productos, loja,estoque
WHERE productos.id_loja=loja.cod_loja AND
productos.id_loja=estoque.id_loja AND loja.cod_loja = 1;
```

```
SELECT * FROM productos, loja,estoque
WHERE productos.id_loja=loja.cod_loja AND productos.id_loja=estoque.id_loja AND loja.cod_loja = 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 productos
RIGHT JOIN estoque
ON productos.cod_prod = estoque.id_prod;
```

cod_prod	descricao_produto	data_inclusao	preco_produto	id_loja	qtd_prod	id_prod	id_loja
1	boneca	2010-12-30	1000.000	1	100.000	1	1
2	camisa verde	2010-12-28	60.000	3	12.000	2	3
170	Leite condensado mococa	2010-12-30	95.400	2	35.000	170	2

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 *
FROM productos
LEFT JOIN estoque
ON productos.cod_prod = estoque.id_prod;
```

cod_prod	descricao_produto	data_inclusao	preco_produto	id_loja	qtd_prod	id_prod	id_loja
1	boneca	2010-12-30	1000.000	1	100.000	1	1
2	camisa verde	2010-12-28	60.000	3	12.000	2	3
170	Leite condensado mococa	2010-12-30	95.400	2	35.000	170	2
3	camisa azul	2010-11-28	90.000	3	NULL	NULL	NULL
4	cavalinho madeira	2011-11-12	1500.000	1	NULL	NULL	NULL
5	jeans	2009-01-02	250.000	4	NULL	NULL	NULL