**Atividade de banco de dados relacional Mysql.**

**Atividade 3**

Crie um banco de dados para um serviço de farmácia de uma empresa, o nome do banco deverá ter o seguinte nome db\_farmacia\_do\_bem, onde o sistema trabalhará com as informações dos produtos desta empresa.

O sistema trabalhará com 2 tabelas tb\_produto e tb\_categoria.

**siga exatamente esse passo a passo:**

Crie uma tabela de categorias utilizando a habilidade de abstração e determine 3 atributos relevantes do tb\_categoria para se trabalhar com o serviço desta farmacia.

Crie uma tabela de tb\_produto e utilizando a habilidade de abstração e determine 5 atributos relevantes dos tb\_produto para se trabalhar com o serviço deste farmacia**(não esqueça de criar a foreign key de tb\_categoria** **nesta tabela)**.

Popule esta tabela Categoria com até 5 dados.

Popule esta tabela Produto com até 8 dados.

Faça um select que retorne os Produtos com o valor maior do que 50 reais.

Faça um select trazendo os Produtos com valor entre 3 e 60 reais.

Faça um select utilizando LIKE buscando os Produtos com a letra B.

Faça um um select com Inner join entre tabela categoria e produto.

Faça um select onde traga todos os Produtos de uma categoria específica (exemplo todos os produtos que são cosméticos).

salve as querys para cada uma dos requisitos o exercício em um arquivo .SQL ou texto e coloque no seu GitHuB pessoal e compartilhe esta atividade.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| CRIAR BANCO DE DADOS db\_farmacia\_do\_bem ; |  |  |
|  |  |  |
|  |  | USE db\_farmacia\_do\_bem; |
|  |  |  |
|  |  | CRIAR TABELA tb\_categoria ( |
|  |  | cl\_id\_categoria INT AUTO\_INCREMENT, |
|  |  | cl\_type\_categoria VARCHAR ( 255 ) UNIQUE NOT NULL , |
|  |  | cl\_is\_promocao BOOLEAN NOT NULL , |
|  |  | cl\_is\_analgesico\_categoria BOOLEAN NOT NULL , |
|  |  | CHAVE PRIMÁRIA (cl\_id\_categoria) |
|  |  | ); |
|  |  | CRIAR TABELA tb\_produto ( |
|  |  | cl\_id\_produto INT AUTO\_INCREMENT, |
|  |  | cl\_marca\_produto VARCHAR ( 255 ) UNIQUE NOT NULL , |
|  |  | cl\_preco\_produto DECIMAL ( 10 , 2 ) NÃO NULO , |
|  |  | cl\_validade\_produto DATA NÃO NULO , |
|  |  | fk\_cl\_id\_categoria INT , |
|  |  | PRIMARY KEY (cl\_id\_produto), |
|  |  | CONSTRAINT cl\_id\_categoria FOREIGN KEY (fk\_cl\_id\_categoria) |
|  |  | REFERÊNCIAS tb\_categoria (cl\_id\_categoria) |
|  |  | ); |
|  |  |  |
|  |  | INSERT INTO tb\_categoria (cl\_type\_categoria, cl\_is\_promocao, cl\_is\_analgesico\_categoria) |
|  |  | VALORES ( " Remédio " , VERDADEIRO, VERDADEIRO), |
|  |  | ( " Creme " , FALSO, VERDADEIRO), |
|  |  | ( " Remédio-Controlado " , VERDADEIRO, FALSO), |
|  |  | ( " Preservativo " , FALSO, FALSO), |
|  |  | ( " Desodorante " , FALSO, FALSO); |
|  |  |  |
|  |  | INSERT INTO tb\_produto (cl\_marca\_produto, cl\_preco\_produto, cl\_validade\_produto, fk\_cl\_id\_categoria) |
|  |  | VALORES ( " Pôr-do-sol " , 55 , " 2022-06-15 " , 2 ), |
|  |  | ( " Johnson & Johnson " , 20 , " 2026-05-14 " , 2 ), |
|  |  | ( " Dipirona " , 5 , " 2021-08-19 " , 1 ), |
|  |  | ( " Jontex " , 8 , " 2022-12-23 " , 4 ); |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | SELECT \* FROM tb\_produto; |
|  |  | SELECT \* FROM tb\_categoria; |
|  |  |  |
|  |  | SELECT \* FROM tb\_produto WHERE cl\_preco\_produto > 50 ; |
|  |  |  |
|  |  | SELECT \* FROM tb\_produto WHERE cl\_preco\_produto > = 3 AND cl\_preco\_produto <= 60 ; |
|  |  |  |
|  |  | SELECT \* FROM tb\_produto WHERE cl\_marca\_produto LIKE " % b% " ; |
|  |  |  |
|  |  | SELECT tb\_produto . cl\_marca\_produto , tb\_produto . cl\_preco\_produto , tb\_categoria . cl\_type\_categoria |
|  |  | FROM tb\_produto |
|  |  | INNER JOIN tb\_categoria ON tb\_categoria . cl\_id\_categoria = tb\_produto . fk\_cl\_id\_categoria |
|  |  | ONDE tb\_categoria . cl\_type\_categoria LIKE " % Creme% " ; |
|  |  |  |
|  |  | SELECT tb\_produto . cl\_marca\_produto , tb\_produto . cl\_preco\_produto , tb\_categoria . cl\_is\_analgesico\_categoria |
|  |  | FROM tb\_produto |
|  |  | INNER JOIN tb\_categoria ON tb\_categoria . cl\_id\_categoria = tb\_produto . fk\_cl\_id\_categoria |
|  |  | ONDE tb\_categoria . cl\_is\_analgesico\_categoria LIKE FALSE |
|  |  |  |
|  |  |  |
|  |  |  |

**Atividade 4**

Crie um banco de dados para um serviço de um açougue, o nome do banco deverá ter o seguinte nome db\_cidade\_das\_carnes, onde o sistema trabalhará com as informações dos produtos desta empresa.

O sistema trabalhará com 2 tabelas tb\_produto e tb\_categoria.

**siga exatamente esse passo a passo:**

Crie uma tabela de categorias utilizando a habilidade de abstração e determine 3 atributos relevantes do tb\_categoria para se trabalhar com o serviço desse açougue.

Crie uma tabela de tb\_produto e utilizando a habilidade de abstração e determine 5 atributos relevantes dos tb\_produto para se trabalhar com o serviço desse açougue**(não esqueça de criar a foreign key de tb\_categoria** **nesta tabela)**.

Popule esta tabela Categoria com até 5 dados.

Popule esta tabela Produto com até 8 dados.

Faça um select que retorne os Produtos com o valor maior do que 50 reais.

Faça um select trazendo os Produtos com valor entre 3 e 60 reais.

Faça um select utilizando LIKE buscando os Produtos com a letra C.

Faça um um select com Inner join entre tabela categoria e produto.

Faça um select onde traga todos os Produtos de uma categoria específica (exemplo todos os produtos que são cosméticos).

salve as querys para cada uma dos requisitos o exercício em um arquivo .SQL ou texto e coloque no seu GitHuB pessoal e compartilhe esta atividade.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| CREATE DATABASE db\_cidade\_das\_carnes ; |  |  |
|  |  |  |
|  |  | USE db\_cidade\_das\_carnes; |
|  |  |  |
|  |  | CRIAR TABELA tb\_categoria ( |
|  |  | cl\_id\_categoria INT AUTO\_INCREMENT, |
|  |  | cl\_type\_categoria VARCHAR ( 255 ) UNIQUE NOT NULL , |
|  |  | cl\_promocao\_categoria BOOLEAN NOT NULL , |
|  |  | cl\_is\_carne\_vermelha\_categoria BOOLEAN NOT NULL , |
|  |  | CHAVE PRIMÁRIA (cl\_id\_categoria) |
|  |  | ); |
|  |  | CRIAR TABELA tb\_produto ( |
|  |  | cl\_id\_produto INT AUTO\_INCREMENT, |
|  |  | cl\_nome\_produto VARCHAR ( 255 ) UNIQUE NOT NULL , |
|  |  | cl\_preco\_produto DECIMAL ( 10 , 2 ) NÃO NULO , |
|  |  | cl\_validade\_produto DATA NÃO NULO , |
|  |  | fk\_cl\_id\_categoria INT , |
|  |  | PRIMARY KEY (cl\_id\_produto), |
|  |  | CONSTRAINT cl\_id\_categoria FOREIGN KEY (fk\_cl\_id\_categoria) |
|  |  | REFERÊNCIAS tb\_categoria (cl\_id\_categoria) |
|  |  | ); |
|  |  |  |
|  |  | INSERT INTO tb\_categoria (cl\_type\_categoria, cl\_promocao\_categoria, cl\_is\_carne\_vermelha\_categoria) |
|  |  | VALORES ( " Carne Bovina " , VERDADEIRO, VERDADEIRO), |
|  |  | ( " Carne de Peixe " , FALSO, FALSO), |
|  |  | ( " Carne de Frango " , VERDADEIRO, FALSO); |
|  |  |  |
|  |  |  |
|  |  | INSERT INTO tb\_produto (cl\_nome\_produto, cl\_preco\_produto, cl\_validade\_produto, fk\_cl\_id\_categoria) |
|  |  | VALORES ( " Bisteca " , 55 , " 2022-07-15 " , 1 ), |
|  |  | ( " Patinho " , 40 , " 2026-05-14 " , 1 ), |
|  |  | ( " Costela " , 25 , " 2021-08-19 " , 1 ), |
|  |  | ( " Tilapia " , 25 , " 2021-08-19 " , 2 ), |
|  |  | ( " Peito " , 20 , " 2022-12-23 " , 3 ); |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | SELECT \* FROM tb\_produto; |
|  |  | SELECT \* FROM tb\_categoria; |
|  |  |  |
|  |  | SELECT \* FROM tb\_produto WHERE cl\_preco\_produto > 50 ; |
|  |  |  |
|  |  | SELECT \* FROM tb\_produto WHERE cl\_preco\_produto > = 3 AND cl\_preco\_produto <= 60 ; |
|  |  |  |
|  |  | SELECT \* FROM tb\_produto WHERE cl\_nome\_produto LIKE " % b% " ; |
|  |  |  |
|  |  | SELECT tb\_produto . cl\_nome\_produto , tb\_produto . cl\_preco\_produto , tb\_categoria . cl\_type\_categoria |
|  |  | FROM tb\_produto |
|  |  | INNER JOIN tb\_categoria ON tb\_categoria . cl\_id\_categoria = tb\_produto . fk\_cl\_id\_categoria |
|  |  | ONDE tb\_categoria . cl\_type\_categoria LIKE " % Carne Bovina% " ; |
|  |  |  |
|  |  | SELECT tb\_produto . cl\_nome\_produto , tb\_produto . cl\_preco\_produto , tb\_categoria . cl\_promocao\_categoria |
|  |  | FROM tb\_produto |
|  |  | INNER JOIN tb\_categoria ON tb\_categoria . cl\_id\_categoria = tb\_produto . fk\_cl\_id\_categoria |
|  |  | ONDE tb\_categoria . cl\_promocao\_categoria LIKE FALSE |