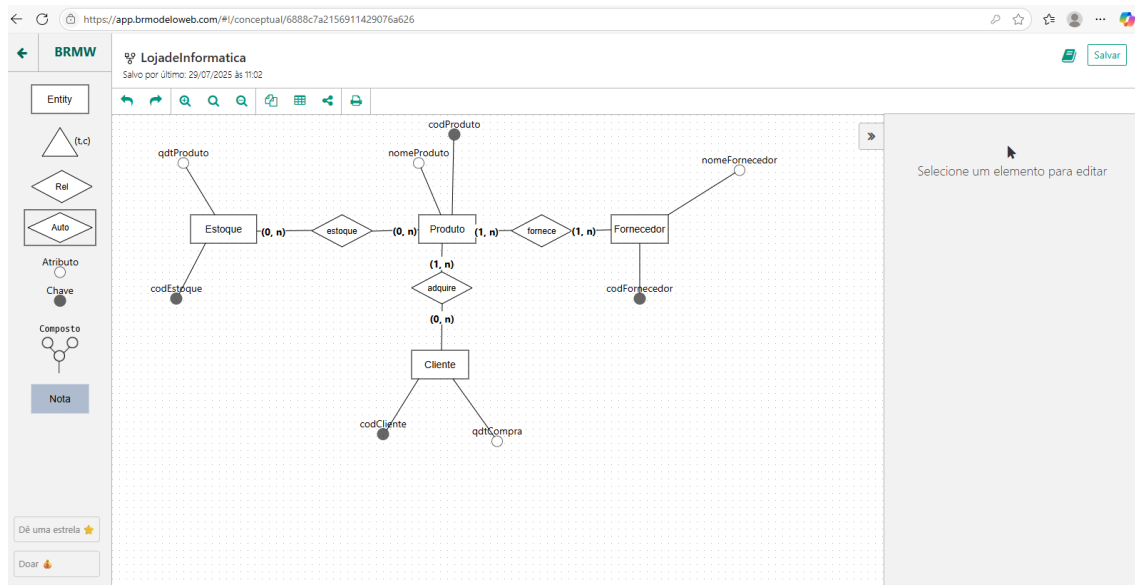


Trabalho de Banco de Dados - Projeto – Parte I

Objetivo: Você foi designado para desenvolver um banco de dados MySQL para uma instituição de: FriStorm Informática. O projeto deve atender às necessidades identificadas por meio de pesquisas internas com os colaboradores da empresa. Para isso, será necessário criar os seguintes modelos de banco de dados:

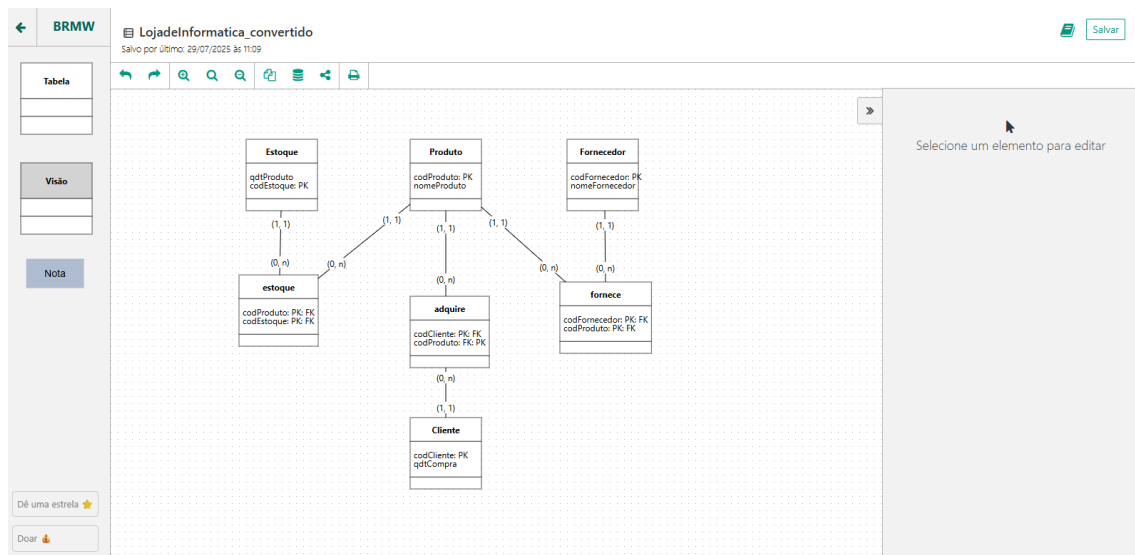
1. Modelo Conceitual: o Criar um diagrama com no mínimo 4 entidades. o Definir corretamente os relacionamentos entre as entidades. o Indicar as cardinalidades existentes. o Apresentar um documento descritivo explicando as escolhas feitas no modelo.



O Modelo Conceitual representa a visão de alto nível do sistema de banco de dados da **Fristorm Informática**, identificando as entidades principais, seus atributos e os relacionamentos entre elas.

Melhor visualização acesse: https://github.com/r-milioli/avaliacao-banco-de-dados/blob/main/FRISTORM_INFORMATICA/DOCUMENTACAO/01_Modelo_Conceitual.md

2. Modelo Lógico: o Criado com base no Modelo Conceitual, com a transformação das entidades e relacionamentos em tabelas e colunas. o Definir chaves primárias e chaves estrangeiras. o O modelo lógico deve ser apresentado de forma clara e organizada.



O Modelo Lógico representa a transformação do Modelo Conceitual em estruturas de dados relacionais, definindo tabelas, colunas, chaves primárias e estrangeiras para implementação no SGBD.

Transformação do Modelo Conceitual

Princípio de Transformação:

- Entidades → Tabelas
- Atributos → Colunas
- Relacionamentos N:N → Tabelas de Junção
- Chaves Primárias → PK (Primary Key)
- Relacionamentos → FK (Foreign Key)

Melhor visualização acesse: https://github.com/r-milioli/avaliacao-banco-de-dados/blob/main/FRISTORM_INFORMATICA/DOCUMENTACAO/02_Modelo_Logico.md

3. Modelo Físico: o O modelo físico será implementado diretamente no MySQL com base no modelo lógico. o Cada tabela deve conter pelo menos 10 registros. o Respeitar as definições de integridade referencial, como chaves primárias e estrangeiras. o Criar um arquivo SQL com os comandos para a criação e inserção de dados no banco.

Melhor visualização acesse: https://github.com/r-milioli/avaliacao-banco-de-dados/blob/main/FRISTORM_INFORMATICA/BANCO_DADOS_FOLDER/banco_dados_fristorm.sql

```
-- =====

-- BANCO DE DADOS - FRISTORM INFORMÁTICA

-- Modelo Físico - MySQL

-- =====

-- Criação do banco de dados

CREATE DATABASE IF NOT EXISTS fristorm_informatica;

USE fristorm_informatica;

-- =====

-- CRIAÇÃO DAS TABELAS

-- =====

-- Tabela Estoque

CREATE TABLE Estoque (

    codEstoque INT PRIMARY KEY,

    qdtProduto INT NOT NULL

);

-- Tabela Produto

CREATE TABLE Produto (

    codProduto INT PRIMARY KEY,

    nomeProduto VARCHAR(100) NOT NULL

);

-- Tabela Fornecedor

CREATE TABLE Fornecedor (

    codFornecedor INT PRIMARY KEY,

    nomeFornecedor VARCHAR(100) NOT NULL

);
```

-- Tabela Cliente

```
CREATE TABLE Cliente (  
    codCliente INT PRIMARY KEY,  
    qdtCompra INT NOT NULL  
);
```

-- Tabela de junção estoque (Produto x Estoque)

```
CREATE TABLE estoque (  
    codProduto INT,  
    codEstoque INT,  
    PRIMARY KEY (codProduto, codEstoque),  
    FOREIGN KEY (codProduto) REFERENCES Produto(codProduto),  
    FOREIGN KEY (codEstoque) REFERENCES Estoque(codEstoque)  
);
```

-- Tabela de junção adquire (Produto x Cliente)

```
CREATE TABLE adquire (  
    codCliente INT,  
    codProduto INT,  
    PRIMARY KEY (codCliente, codProduto),  
    FOREIGN KEY (codCliente) REFERENCES Cliente(codCliente),  
    FOREIGN KEY (codProduto) REFERENCES Produto(codProduto)  
);
```

-- Tabela de junção fornece (Produto x Fornecedor)

```
CREATE TABLE fornece (  
    codFornecedor INT,  
    codProduto INT,  
    PRIMARY KEY (codFornecedor, codProduto),  
    FOREIGN KEY (codFornecedor) REFERENCES Fornecedor(codFornecedor),  
    FOREIGN KEY (codProduto) REFERENCES Produto(codProduto));
```

```
-- =====  
-- INSERÇÃO DE DADOS (10+ registros por tabela)  
-- =====
```

```
-- Inserindo dados na tabela Estoque
```

```
INSERT INTO Estoque (codEstoque, qdtProduto) VALUES
```

```
(1, 150),  
(2, 200),  
(3, 75),  
(4, 300),  
(5, 120),  
(6, 80),  
(7, 250),  
(8, 90),  
(9, 180),  
(10, 110);
```

```
-- Inserindo dados na tabela Produto
```

```
INSERT INTO Produto (codProduto, nomeProduto) VALUES
```

```
(1, 'Notebook Dell Inspiron'),  
(2, 'Mouse Wireless Logitech'),  
(3, 'Teclado Mecânico RGB'),  
(4, 'Monitor LG 24\"'),  
(5, 'Impressora HP LaserJet'),  
(6, 'Webcam HD 1080p'),  
(7, 'Headset Gamer'),  
(8, 'SSD 500GB Kingston'),  
(9, 'Memória RAM 8GB'),  
(10, 'Fonte 500W Corsair'),  
(11, 'Placa de Vídeo GTX 1660'),  
(12, 'Processador Intel i5');
```

-- Inserindo dados na tabela Fornecedor

INSERT INTO Fornecedor (codFornecedor, nomeFornecedor) VALUES

(1, 'Dell Brasil'),

(2, 'Logitech'),

(3, 'LG Electronics'),

(4, 'HP Brasil'),

(5, 'Kingston Technology'),

(6, 'Corsair'),

(7, 'NVIDIA'),

(8, 'Intel Brasil'),

(9, 'Samsung'),

(10, 'Western Digital'),

(11, 'ASUS'),

(12, 'MSI');

-- Inserindo dados na tabela Cliente

INSERT INTO Cliente (codCliente, qdtCompra) VALUES

(1, 5),

(2, 3),

(3, 8),

(4, 2),

(5, 6),

(6, 4),

(7, 7),

(8, 1),

(9, 9),

(10, 3),

(11, 5),

(12, 2);

-- Inserindo dados na tabela estoque (relacionamento Produto x Estoque)

INSERT INTO estoque (codProduto, codEstoque) VALUES

(1, 1), (1, 2), (1, 3),
(2, 1), (2, 4), (2, 5),
(3, 2), (3, 6), (3, 7),
(4, 3), (4, 8), (4, 9),
(5, 4), (5, 10), (5, 1),
(6, 5), (6, 2), (6, 3),
(7, 6), (7, 4), (7, 5),
(8, 7), (8, 8), (8, 9),
(9, 8), (9, 10), (9, 1),
(10, 9), (10, 2), (10, 3),
(11, 10), (11, 4), (11, 5),
(12, 1), (12, 6), (12, 7);

-- Inserindo dados na tabela adquire (relacionamento Produto x Cliente)

INSERT INTO adquire (codCliente, codProduto) VALUES

(1, 1), (1, 2), (1, 3),
(2, 2), (2, 4), (2, 5),
(3, 3), (3, 6), (3, 7),
(4, 4), (4, 8), (4, 9),
(5, 5), (5, 10), (5, 11),
(6, 6), (6, 12), (6, 1),
(7, 7), (7, 2), (7, 3),
(8, 8), (8, 4), (8, 5),
(9, 9), (9, 6), (9, 7),
(10, 10), (10, 8), (10, 9),
(11, 11), (11, 10), (11, 12),
(12, 12), (12, 1), (12, 2);

-- Inserindo dados na tabela fornece (relacionamento Produto x Fornecedor)

INSERT INTO fornece (codFornecedor, codProduto) VALUES

(1, 1), (1, 2), (1, 3),
(2, 2), (2, 4), (2, 5),
(3, 4), (3, 6), (3, 7),
(4, 5), (4, 8), (4, 9),
(5, 8), (5, 9), (5, 10),
(6, 10), (6, 11), (6, 12),
(7, 11), (7, 1), (7, 2),
(8, 12), (8, 3), (8, 4),
(9, 1), (9, 5), (9, 6),
(10, 2), (10, 7), (10, 8),
(11, 3), (11, 9), (11, 10),
(12, 4), (12, 11), (12, 12);

-- =====

-- VERIFICAÇÃO DOS DADOS INSERIDOS

-- =====

-- Contagem de registros em cada tabela

SELECT 'Estoque' as Tabela, COUNT(*) as Total_Registros FROM Estoque

UNION ALL

SELECT 'Produto', COUNT(*) FROM Produto

UNION ALL

SELECT 'Fornecedor', COUNT(*) FROM Fornecedor

UNION ALL

SELECT 'Cliente', COUNT(*) FROM Cliente

UNION ALL

SELECT 'estoque (junção)', COUNT(*) FROM estoque

UNION ALL

SELECT 'adquire (junção)', COUNT(*) FROM adquire

UNION ALL


```
SELECT 'fornece (junção)', COUNT(*) FROM fornece;
```