

DOCUMENTAÇÃO DE BANCO DE DADOS

Documentação elaborada para composição
da nota da Disciplina de Projeto de Banco
de Dados, orientado pelo prof^a Tacyana
Batista.

Alunos: Rodrigo Antonio e Gustavo Rodrigues

Sumário

Minimundo

MER

DER

Scripts DDL

Scripts DML

Minimundo:

Com o intuito de gerenciar e expandir o comércio na cafeteria FlowCoffee, o dono pediu para que um desenvolvedor o ajudasse na elaboração de um banco de dados cumprindo alguns requisitos. Um cliente pode realizar vários pedidos e cada pedido pode ser associado a um funcionário responsável pelo atendimento e pode ter um ou mais produtos, a entrega (delivery) de um pedido só pode ser realizada por um único funcionário.

Ao entrar no estabelecimento o cliente deve se dirigir a um dos caixas disponíveis para iniciar a realização do seu pedido.

O funcionário que realiza os pedidos ao iniciar o atendimento deve perguntar se o cliente possui cadastro e se o cliente possuir ele continua normalmente, caso contrário ele deve criar um cadastro para ele com as seguintes informações: nome, cpf e endereço caso queira delivery.

O cliente pode escolher se deseja consumir no local ou se prefere a entrega via delivery, se tiver informado o endereço no cadastro, onde um funcionário será designado para levar o pedido até o endereço do cliente. Os clientes que optam por consumir no local aguardam no balcão, onde seus pedidos são chamados pela ordem do horário em que foram realizados, ou seja, quem fez o pedido primeiro tem prioridade na entrega do pedido em relação aos pedidos realizados posteriormente.

Os caixas da loja têm acesso aos produtos que contém todos os pratos e bebidas disponíveis bem como o gerenciamento de estoque e seus respectivos valores para facilitar o processo de realização dos pedidos. Cada pedido contém informações detalhadas sobre os produtos escolhidos pelo cliente, o valor total e o horário em que foi realizado.

Os funcionários podem ser de diferentes tipos: atendentes de caixas, cozinheiros e entregadores responsáveis pela entrega dos pedidos via delivery.

O sistema deve permitir consultar o histórico de pedidos de cada cliente, além de possibilitar o controle dos funcionários responsáveis pelos atendimentos e entregas, garantindo o acompanhamento completo do ciclo do pedido desde o registro até a entrega final.

MER:

Cliente (id_clientePK, nome_cliente, endereço, cpf_cliente)

Funcionários (id_funcionárioPK, nome_funcionário, cargo, cpf_funcionário)

Produtos (id_produtoPK, nome_produto, valor_produto, estoque, tipo_produto)

Pedidos (id_pedidoPK, id_clienteFK, id_funcionárioFK, hora_pedido, valor_pedido, observacao, modalidade, status)

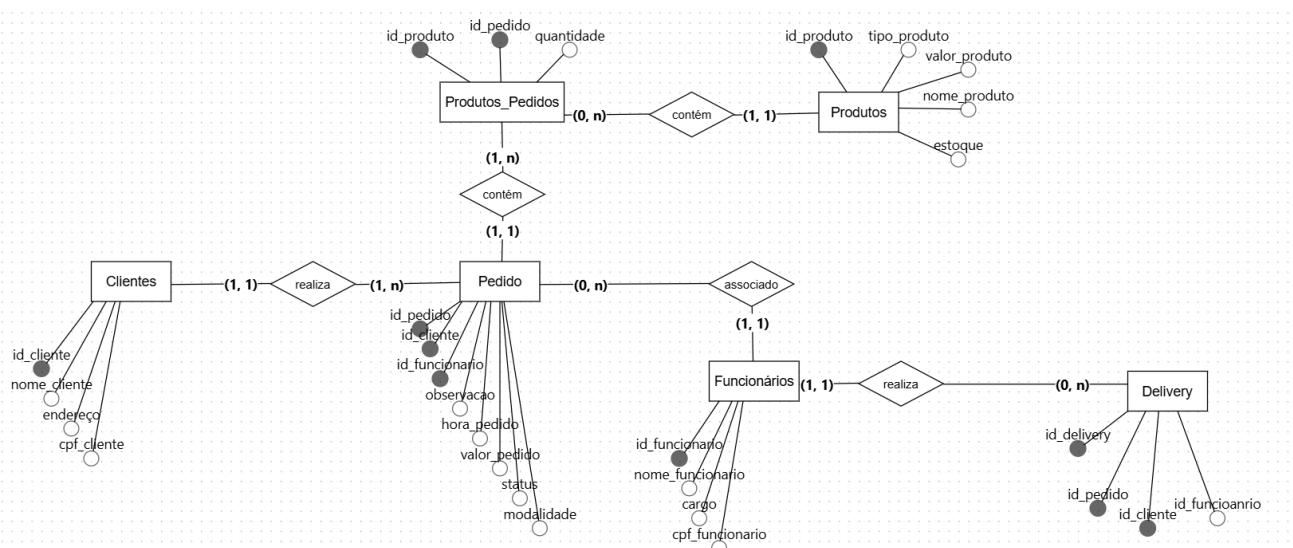
Delivery (id_deliveryPK, id_pedidoFK, id_funcionarioFK, id_clienteFK)

Produtos_Pedidos (id_produtoFK, id_pedidoFK, quantidade)

RELACIONAMENTOS E CARDINALIDADES:

- (1,1) Cliente -- realiza -- pedido (1,n)
- (0,n) Pedido -- é associado -- funcionário (1,1)
- (0,n) Produtos_Pedidos -- Contem -- Produtos (1,1)
- (1,1) Pedidos -- Contém -- Produtos_Pedidos (1,n)
- (1,1) Funcionário -- realiza -- delivery (0,n)

DER:



SCRIPTS DDL:

```

create table CLIENTES(
    id_cliente serial primary key,
    nome_cliente varchar(255) not null,
    cpf_cliente char(11) unique not null,
    endereco varchar(255)
);

create table FUNCIONARIOS(
    id_funcionario serial primary key,
    nome_funcionario varchar(255) not null,
    cpf_funcionario char(11) not null unique,
    cargo varchar(255) not null
);
create table PRODUTOS(
    id_produto serial primary key,
    nome_produto varchar(255) not null,
    estoque integer not null,
    valor_produto decimal(10,2) not null,
    tipo_produto varchar(255) not null
);
create table PEDIDOS(
    id_pedido serial primary key,
    id_cliente integer not null,
    foreign key (id_cliente) references CLIENTES (id_cliente),
    id_funcionario integer not null,
    foreign key (id_funcionario) references FUNCIONARIOS (id_funcionario),
    observacao text,
    hora_pedido timestamp not null,
    valor_pedido decimal(10,2) not null,
    modalidade char(1) not null,
    status char(1) not null
);
create table DELIVERY(
    id_delivery serial primary key,
    id_pedido integer not null,
    foreign key (id_pedido) references PEDIDOS (id_pedido),
    id_cliente integer not null,
    foreign key (id_cliente) references CLIENTES (id_cliente),
    id_funcionario integer not null,
    foreign key (id_funcionario) references FUNCIONARIOS (id_funcionario)
);
create table PRODUTOS_PEDIDOS(
    id_pedido integer not null,
    id_produto integer not null,
    primary key (id_pedido, id_produto),
    quantidade integer not null default 1,
    foreign key (id_pedido) references PEDIDOS (id_pedido),
    foreign key (id_produto) references PRODUTOS (id_produto)
);

```

SCRIPTS DML:

```
insert into CLIENTES (nome_cliente, cpf_cliente, endereco) values
('Rodrigo Antonio', '11565407474', 'Condomínio Melville'),
('Gustavo Rodrigues', '00187612343', 'Edifício Mirante Club Stratégia');
```

```
insert into FUNCIONARIOS (nome_funcionario, cpf_funcionario, cargo) values
('Kleber Ronaldo', '98755436422', 'Chef de Cozinha'),
('Nico Gonzales', '09987656788', 'Caixa'),
('Renato Mosquito', '11345477765', 'Entregador');
```

```
insert into PRODUTOS (nome_produto, tipo_produto, estoque, valor_produto) values
('Café Com Leite', 'Bebida', 11, 12.90),
('Cappuccino', 'Bebida', 18, 15.90),
('Café Espresso', 'Bebida', 5, 8.90),
('Coca-Cola Lata', 'Bebida', 22, 5.90),
('Pão de Queijo', 'Alimentos', 10, 1.90),
('Misto Quente', 'Alimentos', 5, 9.90),
('Coxinha de Frango', 'Alimentos', 13, 6.90),
('Sorvete', 'Sobremesa', 20, 4.50),
('Picolé', 'Sobremesa', 12, 6.50);
```

```
insert into PEDIDOS (id_cliente, id_funcionario, observacao, hora_pedido, valor_pedido,
modalidade, status) values
(1, 2, 'Café Sem Açúcar Por Favor', '2025-10-10 10:32:05', 22.60, 'L', 'P'),
(2, 2, 'Sorvete de Uva', '2025-10-10 12:02:54', 17.30, 'E', 'C');
```

```
insert into PRODUTOS_PEDIDOS (id_pedido, id_produto, quantidade) values
(1, 3, 1),
(1, 5, 2),
(1, 6, 1),
(2, 4, 1),
(2, 7, 1),
(2, 8, 1);
```

```
insert into DELIVERY (id_pedido, id_cliente, id_funcionario) values
(1, 1, 3),
(2, 2, 3);
```