

Universidade Federal de Uberlândia

Felipe Alves Belisário
Breno Caldeira Nascimento
Adiel Pereira Prado

Projeto

Pizzaria com Entretenimento

Curso: Bacharelado em Ciência da Computação
Disciplina: Sistema de Banco de Dados (GBC043)
Professora: Profa. Maria Camila Nardini Barioni
Data de entrega: 03/05/2019

ÍNDICE

1 Especificação do Problema / Esquema Conceitual.....	3
2 Esquema Relacional	7

1 – Especificação do Problema / Esquema conceitual

A) Os tópicos a seguir descrevem o funcionamento do banco de dados utilizado no aplicativo.

B) Os usuários são identificados de forma única. Logo, usamos o CPF deste como chave identificadora. Também utilizamos data de registro, nome e endereço para armazenar no sistema.

C) Uma das opções que o usuário possui ao se cadastrar é de ser um Dono de Negócio. Para essa disjunção, registramos sua conta do LinkedIn.

D) Um dono de negócio pode possuir várias pizzarias, porém, uma pizzaria terá apenas um dono. Modelamos essa situação com uma cardinalidade 1 para n, entre “Dono de negócio”, “Possui” e “Pizzaria”.

E) A entidade “Pizzaria” possui os seguintes atributos: CEP, endereço, número de telefone, web site, e horários de funcionamento e nome. Criamos este último atributo e o usamos como chave, visto que é único para cada estabelecimento.

F) Criamos uma entidade fraca “Pizza”, cujos atributos são: nome e preço, onde aquele é uma chave fraca. A entidade é fraca pois duas pizzas podem possuir o mesmo sabor e o mesmo preço, porém, vindas de estabelecimentos diferentes, tornando-as únicas para cada pizzaria.

G) Uma pizza pertence a uma categoria, porém uma categoria possui várias pizzas. Logo, justifica-se a cardinalidade 1 para n, no relacionamento “Pertence” entre “Pizza” e “Categoria”. Este é uma entidade com os seguintes atributos: Descrição, e Código, usado como chave. Além disso, essa recém criada entidade “Categoria” faz uma relação consigo mesma já que dentro de uma categoria podem haver várias subcategorias e uma subcategoria pode estar contida em uma categoria.

H) Criamos uma entidade fraca “Acompanhamento”, que depende de uma pizzaria, seus atributos são: nome, descrição, tipo de acompanhamento e preço. O código é usado como chave para distingui-lo dentro de uma única pizzaria.

I) Como citado no item b), outra opção para o usuário é de ser um “Consumidor Faminto”, uma entidade que possui endereço de entrega como atributo.

J) Criamos uma entidade “Pedido”, com um atributo ID, usado como chave. A relação “Realiza”, entre “Pedido” e “Consumidor Faminto” possui data e horário como atributos. Adicionamos “Horário Posterior” e “Quantidade de Pessoas” como atributos de “Pedido”, para a escolha pessoal do consumidor.

K) Criamos uma relação “Possui”, entre “Pedido” e “Pizza”, com os seguintes atributos: massa, quantidade de molho e borda. Sua cardinalidade é n para m, visto que um pedido pode conter várias pizzas, e várias pizzas podem pertencer a um pedido. Os atributos estão na relação pois eles “modificam” uma pizza padrão já existente, para incluir na realização do pedido.

L) Criamos a relação “Possui” entre “Pedido” e “Acompanhamento”. Ela possui quantidade como atributo. A cardinalidade e o atributo dela possui a mesma justificativa do item anterior.

M) Fizemos uma agregação envolvendo a relação “Possui”, que se encontra entre “Pedido” e “Pizza”, nomeando-a de “Pizza_Pedido”, a mesma deve existir pois a entidade criada “Ingrediente Extra” se trata de apenas um adicional (opcional) para a pizza. Um ingrediente extra pode estar contido a vários pedidos de pizzas e um pedido de pizzas pode conter vários ingredientes extras.

N) Um “Ingrediente Extra” possui um código, nome e preço como atributos. O primeiro é tratado como chave. A existência desses atributos provam este ser uma entidade.

O) A entidade “Pedido com Entretenimento” é uma disjunção de “Pedido”, visto que herda os mesmos atributos. Além disso, possui tipo e duração como atributos.

P) Como o preço é calculado a partir das pizzas escolhidas, ingredientes extras, acompanhamentos e escolha de entretenimento, ele torna-se um atributo derivado de “Pedido”.

Q) Um usuário, ao se cadastrar no sistema, pode ser um animador. Para isso, criamos uma disjunção de “Usuário” chamada “Animador”. Ela possui nome artístico, biografia e preço, para 30 minutos de entretenimento, como atributos.

R) Um “Pedido com Entretenimento” possui apenas um animador. Logo, justifica-se a cardinalidade 1 para n da relação “Atende” entre “Animador” e “Pedido com entretenimento”.

S) Criamos a relação “Trabalha” entre “Animador” e “Pizzaria”. Nela, o animador especifica os dias da semana que ele está disponível, logo, a disponibilidade dele se tornará um atributo multivalorado da relação “Trabalha”.

Extra) Criamos um atributo “Gorjeta” que faz parte da relação “Atende” entre “Pedido com entretenimento” e “Animador” com o intuito de os animadores poderem receber um valor a mais sobre seus serviços vindos do consumidor faminto. E o outro requisito adicional que adicionamos é colocar um atributo chave “Nome” para cada pizzaria para que elas possam ser identificadas pelo seu nome.

2 – Esquema Relacional

Para mapear o modelo ER feito na tarefa 1 seguimos o passo a passo e mapeamos primeiramente as entidades fortes e depois as fracas que não são subclasses, as quais só se foram passadas uma forma de serem implementadas em sala de aula. Logo após isso, foram mapeadas todas as relações 1:n, já que não há nenhuma de 1:1, utilizando o método no qual a entidade com a cardinalidade n recebe a chave estrangeira, que não será chave primária da mesma, vinda da entidade de cardinalidade 1. Depois disso vieram os relacionamentos n:m, onde foi criada uma tabela para cada um deles e neles foram recebidas as chaves estrangeiras vindas das duas entidades as quais faz parte, e estas se tornam também suas chaves primárias.

Além disso também foi mapeado o único atributo multivalorado presente no modelo de acordo com o método passado de se criar uma tabela que possui chave estrangeira vinda da relação na qual faz parte, sendo ela também chave primária da nova tabela criada junto com a própria chave já presente nela. Após, foi a vez das subclasses, em que foi utilizado o método 8A devido à grande quantidade de atributos das superclasses. Por último foi mapeada a agregação a qual se tornou uma tabela separada da do relacionamento que ela se encontra pois os atributos desta última não podem ser repassados para a agregação já que a função dela é apenas tornar os ingredientes extras da pizza opcionais.

OBS: Atributos derivados não são mapeados

