

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS - PUC MINAS
TECNOLOGIA EM BANCO DE DADOS

CARLOS ALEXANDRE KWIATKOWSKI
ANDERSON BOLIVAR NASCIMENTO
GILCICLEIDE DANTAS DE MATOS
RAFAEL DE SOUSA
VANESSA OMACHT SCHNEE

PROJETO E DESENVOLVIMENTO DE SISTEMA DE BANCO DE DADOS
RELACIONAL

BELO HORIZONTE – 2023

1 INTRODUÇÃO

Restaurantes necessitam manter um controle financeiro do fluxo de caixa para a viabilidade do negócio se manter sustentável. Também, para entendimento das estratégias de marketing, é necessário ter registro da quantidade de clientes do restaurante, quais são mais cativos, qual o ticket médio dos pedidos por cliente, etc. A modelagem dos dados e a manutenção destes são imprescindíveis para o bom funcionamento de qualquer restaurante.

No caso, o objetivo do trabalho é um projeto de banco de dados para um sistema de administração de um restaurante de massas online. Esse sistema terá o registro de pedidos, clientes, estoque e transações de créditos e débitos. Poderemos após o preenchimento do banco de dados citado, realizar consultas que trarão insights sobre o desempenho do negócio.

Propõe-se uma especificação de minimundo, análise de requisitos, projeto conceitual, projeto lógico e projeto físico do banco de dados ARO (Administração de Restaurante Online), que em sua versão utiliza como base o modelo relacional e podendo ser implementado em sistemas gerenciadores de banco de dados (SGBD) relacionais comerciais.

2 ESPECIFICAÇÃO DO MINIMUNDO

Essa seção apresenta a descrição textual de minimundo do ARO (v1.0), um banco de dados para um sistema de restaurantes que atende o público on-line no comércio de pratos (massas).

O cliente (client) irá realizar o seu cadastro utilizando o CPF, sendo este o campo de identificação do cliente dentro da base de dados. Além do mais ele poderá realizar o cadastro de vários endereços de entrega de pedidos (address), criando um apelido único para cada um deles.

Já as refeições (meal) serão disponibilizadas através de um serviço A La Carte com cardápio definido pelo estabelecimento. Cada refeição terá um único estoque (meal_stock) contendo informações sobre a disponibilidade da mesma. O estoque como modelado será uma abstração para os insumos de ingredientes de uma refeição. Ao invés de considerar a disponibilidade de cada um dos ingredientes necessários para o preparo da mesma, vamos resumir em um único número que define a quantidade de pratos disponíveis no restaurante.

Quanto aos pedidos (order), estes serão feitos por um cliente e direcionados a um endereço cadastrado pelo mesmo. Antes de se fazer o pedido é feita a verificação da quantidade de refeições disponíveis em estoque. Se a quantidade solicitada pelo cliente da refeição estiver disponível, o pedido é criado, caso contrário, o pedido é cancelado. Depois de feito o pedido, é criada uma transação (transaction) de entrada de dinheiro

com a propriedade *code* sendo *orderId* e o registro dos pratos inclusos no pedido (*order_meal*).

A cada refeição solicitada (*order_meal*) irá deduzir sua venda da quantidade de refeições disponíveis no estoque da refeição (*meal_stock*). A cada atualização do estoque, caso a quantidade disponível for menor que a quantidade ideal de refeições para um certo período, a administração será notificada para criação de uma ordem de estoque (*stock_order*) com a quantidade de refeições solicitadas. Cada ordem de estoque realizada aumentará a quantidade de refeições disponíveis no estoque de refeições e criará uma transação (*transaction*) de saída de dinheiro com a propriedade *code* sendo *stockOrderId*.

As transações (*transactions*) representam todo o fluxo de caixa do restaurante. Existirá as transações com *code* definido para o caso em que há criação automática de uma transação de entrada de dinheiro após um pedido ser feito, como também para o caso em que há criação automática de uma transação de saída de dinheiro após uma ordem de estoque ser feita. Haverá também transações pré-estabelecidas, com *code* nulo. Para essas transações, a fim de se preencher a base de dados com dados próximos da realidade, após pesquisa de valores de mercado, vamos popular as transações com os seguintes dados:

- Caixa inicial do restaurante, montante de investimento no negócio.
- Aluguel do estabelecimento de cozinha cobrado mensalmente.
- Salários de funcionários do estabelecimento.
- Gastos com demais insumos para manutenção do restaurante.

2.1 REQUISITOS FUNCIONAIS

Diferentes grupos de usuários demandarão diferentes operações de manipulação de dados sobre diferentes porções do banco de dados. O grupo **CLIENTE** será responsável por fazer seu cadastro, atualizar suas informações e endereço, e fazer pedidos dentro do aplicativo. O grupo **GERÊNCIA** terá acesso a dados que dizem respeito ao funcionamento do restaurante, como verificar a quantidade de refeições em estoque e o cadastro das mesmas. Já o grupo **ADMINISTRAÇÃO** é responsável por toda a parte financeira, e tem acesso aos dados do negócio que geram insights, como visualização de relatórios débitos e créditos, quantidade de pedidos, refeições mais pedidas, etc. A Tabela 1 apresenta as principais consultas que cada grupo de usuários demandará ao sistema de banco de dados, bem como a frequência esperada de submissão.

Tabela 1: Frequência esperada de consultas por grupo de usuário

	Consulta	Grupo	Frequência
Q001	Cadastro de cliente na base de dados.	Cliente	Baixa
Q002	Cadastro de endereços na base de dados.	Cliente	Baixa
Q003	Cadastro de refeições e seus respectivos estoques na base de dados.	Gerência	Baixa
Q004	Verificar disponibilidade de refeições no estoque.	Gerência	Média
Q005	Criar pedido associado a um cliente, a um endereço e respectivas refeições.	Cliente	Alta
Q006	Atualizar quantidade de refeições em estoque.	Gerência	Média
Q007	Criar transações financeiras de entradas e saídas do fluxo de caixa.	Administração	Alta
Q008	Criar ordem de estoque.	Administração	Média
Q009	Visualizar a soma de todas as receitas geradas por pedidos no período	Administração	Média
Q010	Visualizar a quantidade de pedidos feitos no período.	Administração	Média
Q011	Visualizar a quantidade total de clientes.	Administração	Baixa
Q012	Visualizar a soma dos gastos por categoria: gastos com estoque, gastos gerais.	Administração	Baixa
Q013	Leitura de todos os clientes cadastrados na base de dados.	Administração	Média
Q014	Visualizar refeições mais e menos pedidas por período.	Administração	Média
Q015	Visualizar os 10 clientes que mais compraram, em valor total por período.	Administração	Média

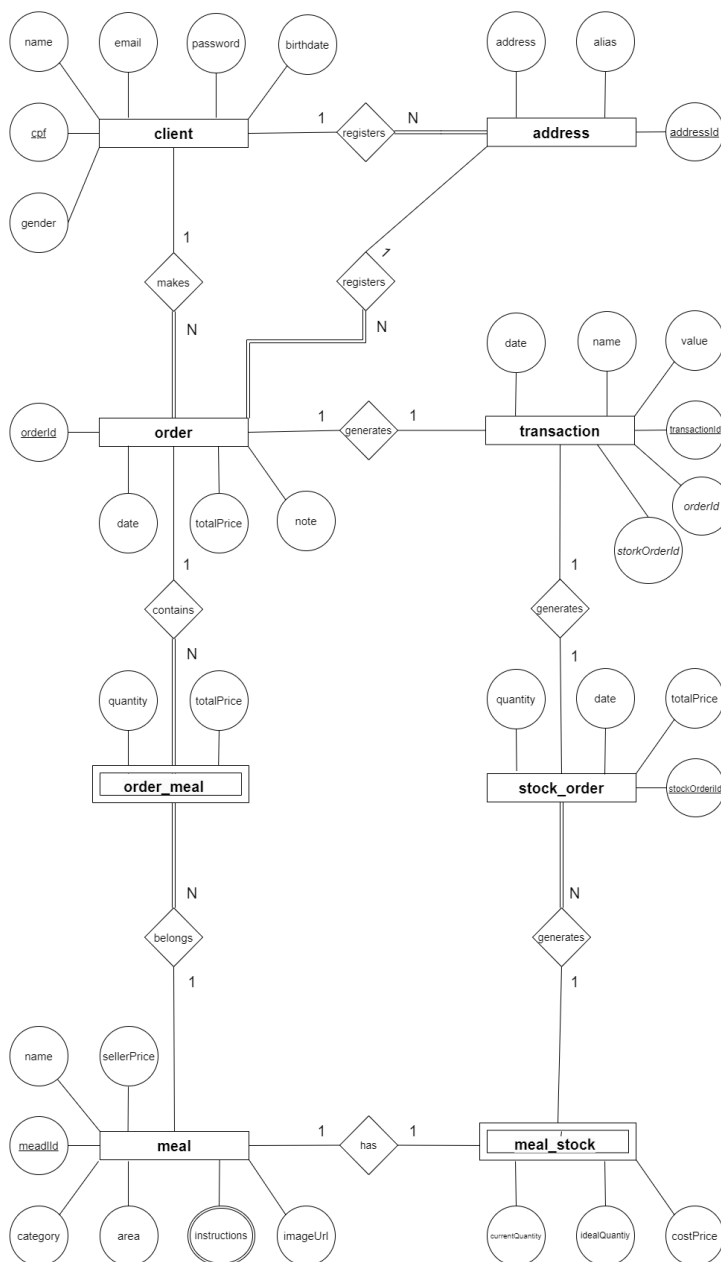
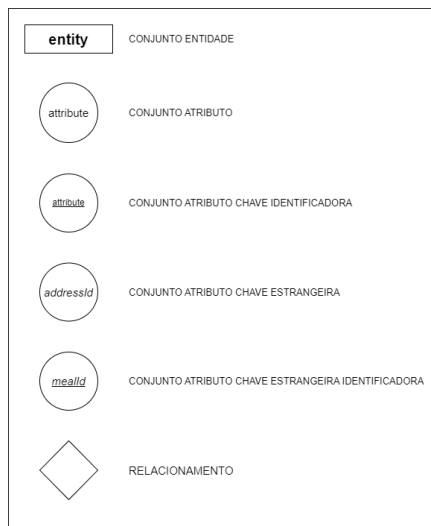
3 PROJETO CONCEITUAL

Essa seção apresenta o projeto conceitual do ARO (v1.0), descrevendo as principais estruturas e restrições conceituais do banco de dados.

Figura 1: Diagrama ER do modelo conceitual do banco de dados ARO (v1.0)

DIAGRAMA DE CHEN

LEGENDA



A Tabela 2 com o objetivo de dar mais detalhes acerca dos elementos descritos no diagrama apresentado na Figura 1.

Tabela 2: Elementos do modelo conceitual do banco de dados ARO

Tipo	Subtipo	Rótulo	Referência	Descrição
------	---------	--------	------------	-----------

Entidade	Forte	Client		Clientes cadastrados no restaurante
Entidade	Forte	Address		Endereços cadastrados por clientes.
Entidade	Forte	Meal		Refeições do restaurante.
Entidade	Forte	Transaction		Movimentações financeiras do restaurante.
Entidade	Forte	Order		Pedidos feitos por clientes.
Entidade	Fraca	Meal_Stock		Estoque de refeições.
Entidade	Fraca	Order_Meal		Refeições inclusas no pedido.
Entidade	Forte	Stock_Order		Pedido de estoque de uma refeição.
Relacionamento	Forte	makes	Client, Order	Um cliente faz vários pedidos
Relacionamento	Forte	registers	Client, Address	Um cliente registra vários endereços
Relacionamento	Forte	registers	Address, Order	Um endereço está registrado em vários pedidos
Relacionamento	Forte	generates	Order, Transaction	Um pedido cria uma transação de entrada de dinheiro
Relacionamento	Forte	generates	Stock_Order, Transaction	Uma ordem de estoque cria uma transação de saída de dinheiro
Relacionamento	Fraco	generates	Meal_Stock, Stock_Order	Um estoque de refeições gera uma ordem de estoque
Relacionamento	Fraco	contains	Order_Meal, Order	Um pedido contém várias refeições
Relacionamento	Fraco	belongs	Order_Meal, Meal	Uma refeição pertence a vários pedidos
Relacionamento	Fraco	has	Meal, Meal_Stock	Uma refeição tem um estoque

Atributo	Simples	name	Client	Nome do cliente
Atributo	Simples	email	Client	Email do cliente
Atributo	Simples	password	Client	Senha do cliente
Atributo	Simples	birthdate	Client	Data de aniversário do cliente
Atributo	Chave	CPF	Client	CPF do cliente
Atributo	Simples	gender	Client	Gênero do cliente
Atributo	Composto	address	Address	Endereço do cliente
Atributo	Simples	alias	Address	Apelido do endereço
Atributo	Chave	addressId	Address	Identificador do endereço
Atributo	Chave	orderId	Order	Identificador do pedido
Atributo	Simples	date	Order	Data do pedido
Atributo	Simples	totalPrice	Order	Preço total do pedido
Atributo	Simples	note	Order	Observações do pedido
Atributo	Simples	date	Transaction	Data da transação
Atributo	Simples	name	Transaction	Nome da transação
Atributo	Simples	value	Transaction	Valor da transação
Atributo	Chave	transactionId	Transaction	Identificador da transação
Atributo	Chave	stockOrderId	Transaction	Transação de um pedido de estoque
Atributo	Chave	orderid	Transaction	Transação de um pedido
Atributo	Simples	TotalPrice	Order_Meal	Preço total da refeição inclusa no pedido
Atributo	Simples	Quantity	Order_Meal	Quantidade da refeição inclusa no pedido
Atributo	Simples	quantity	Stock_Order	Quantidade de uma refeição em uma ordem de estoque

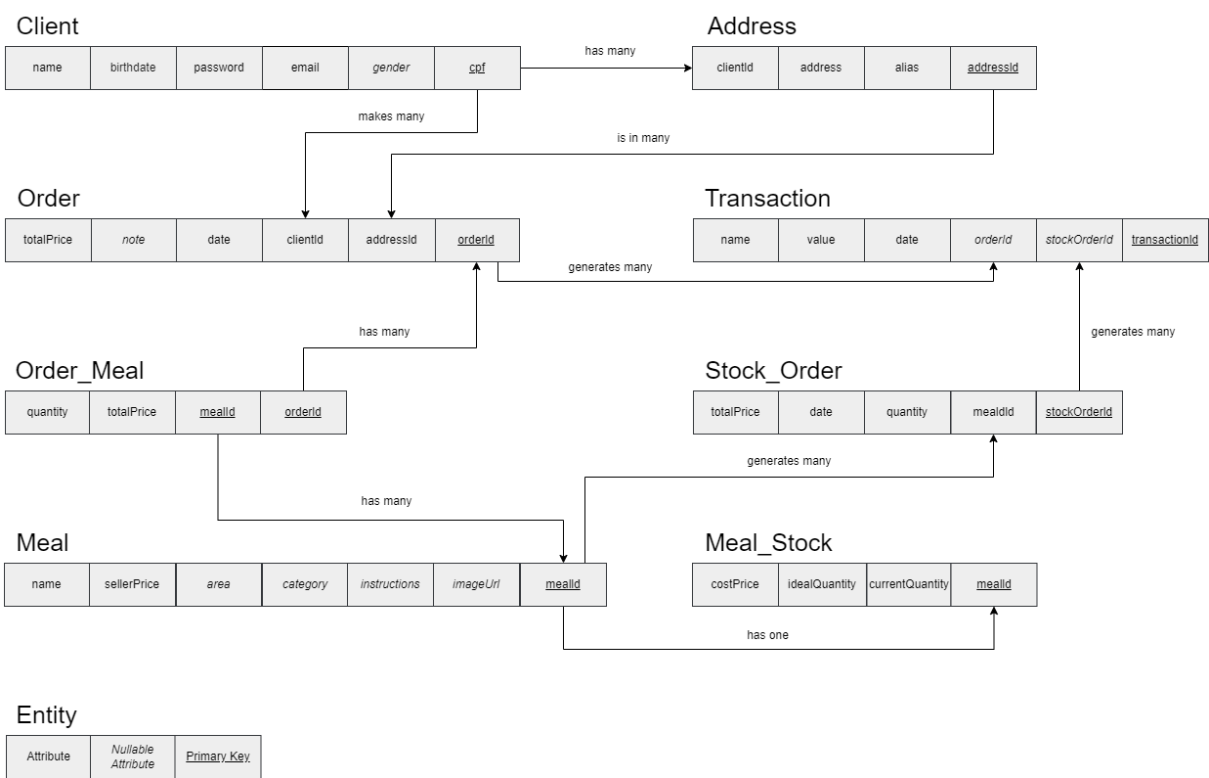
Atributo	Simples	date	Stock_Order	Data da ordem de estoque
Atributo	Derivado	totalPrice	Stock_Order	Preço total da ordem de estoque
Atributo	Chave	stockorderId	Stock_Order	Identificador de uma ordem de estoque
Atributo	Simples	name	Meal	Nome da refeição
Atributo	Chave	mealId	Meal	Identificador da refeição
Atributo	Simples	category	Meal	Categoria da refeição
Atributo	Simples	sellerPrice	Meal	Preço de venda da refeição
Atributo	Simples	area	Meal	Área de origem da refeição
Atributo	Simples	instructions	Meal	Instruções da refeição
Atributo	Simples	imageUrl	Meal	URL da imagem da refeição
Atributo	Simples	currentQuantity	Meal_Stock	Quantidade disponível de estoque de uma refeição
Atributo	Simples	idealQuantity	Meal_Stock	Quantidade ideal de estoque de uma refeição
Atributo	Simples	costPrice	Meal_Stock	Preço de custo de uma refeição
Restrição	Cardinalidade	1-N	Client, Order	Um cliente faz vários pedidos
Restrição	Cardinalidade	1-N	Client, Address	Um cliente registra vários endereços
Restrição	Cardinalidade	1-N	Address, Order	Um endereço está registrado em vários pedidos
Restrição	Cardinalidade	1-1	Order, Transaction	Um pedido cria uma transação de entrada de dinheiro

Restrição	Cardinalidade	1-1	Stock_Order, Transaction	Uma ordem de estoque cria uma transação de saída de dinheiro
Restrição	Cardinalidade	1-N	Meal_Stock, Stock_Order	Um estoque de refeições gera ordem de estoque
Restrição	Cardinalidade	N-N	Order, Meal	Um pedido contém várias refeições
Restrição	Cardinalidade	1-1	Meal, Meal_Stock	Uma refeição tem um estoque

4 PROJETO LÓGICO

Essa seção apresenta o projeto lógico do banco de dados ARO (v1.0), descrevendo as principais estruturas e restrições lógicas baseadas no modelo de implementação relacional. A Figura 2 apresenta o diagrama relacional do banco de dados, mapeado a partir do modelo conceitual descrito no presente relatório.

Figura 2: Diagrama do modelo de implementação relacional do ARO (v1.0)

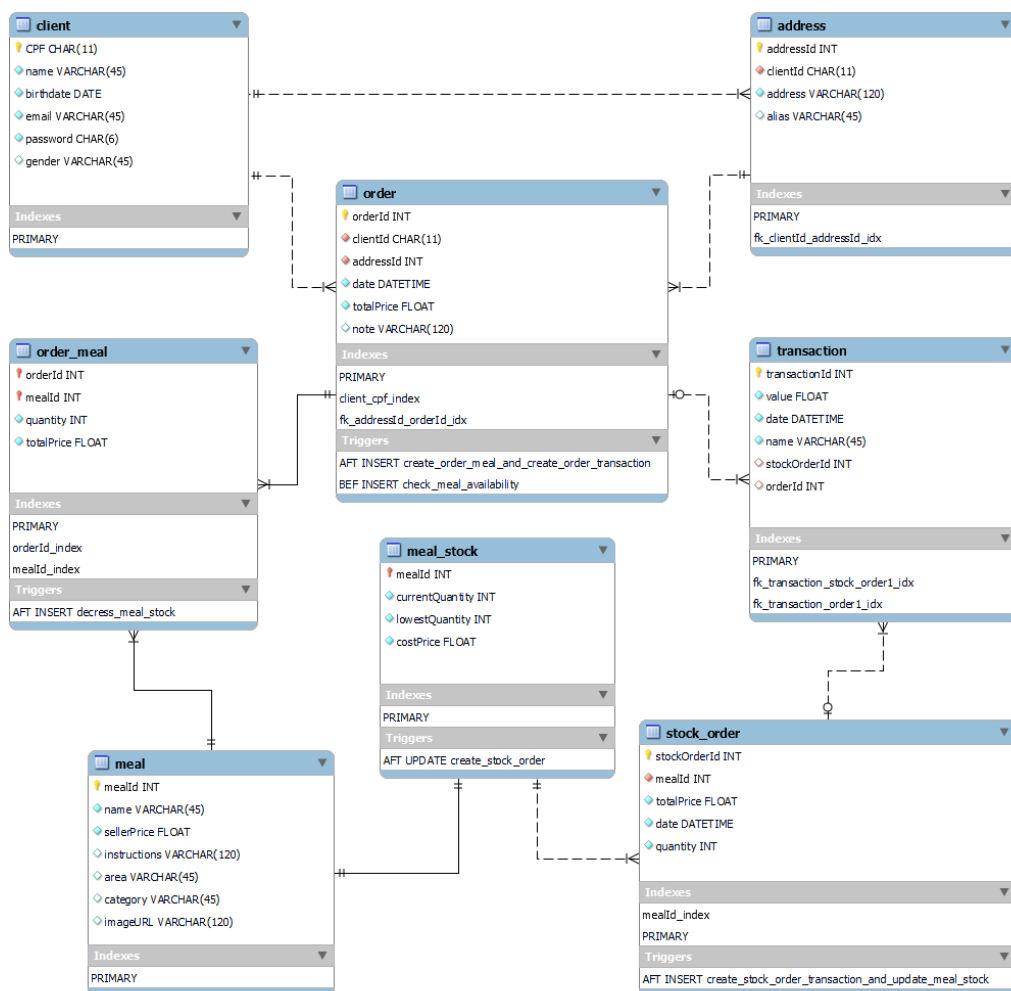


Na Figura 2, estão mapeadas nove relações entre oito entidades, com uma média de aproximadamente sete atributos por relação. Além disso, foram identificadas nove referências entre as relações.

O diagrama relacional apresentado é muito útil para visualizar de maneira simples e compacta as relações, atributos e restrições de chave e integridade referencial presentes no banco de dados, independentemente do SGBD relacional comercial a ser adotado para sua implementação. Cada tipo de entidade ou de relacionamento é representado por um esquema de relação separado descrevendo a estrutura do banco de dados, que pode ser processada por um SGBD.

Além disso, para a implementação do ARO (v1.0) adotaremos a abordagem baseada em SGBD relacional e a solução comercial MySQL. A Figura 3 apresenta o EER do modelo de implementação relacional, incluindo restrições de chave, tipo, nulidade, e integridade referencial.

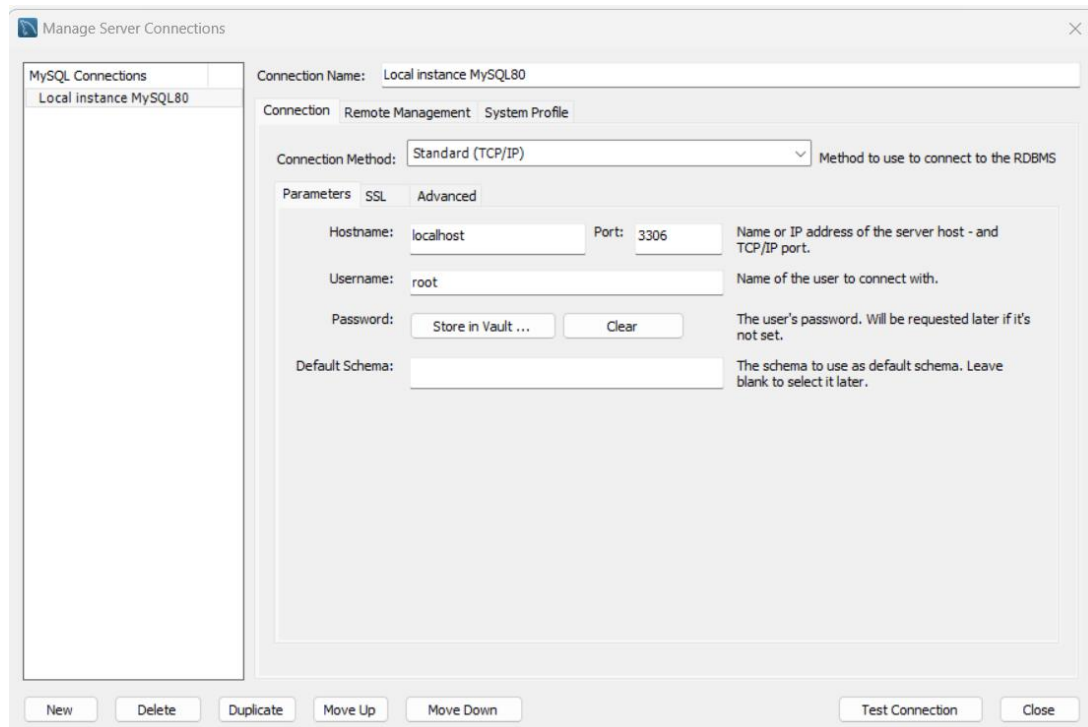
Figura 3: EER do modelo de implementação relacional do ARO (v1.0)



5 PROJETO FÍSICO

O sistema de gestão de base de dados SQL utilizado foi o MySQL. Este sistema é de código aberto, desenvolvido, distribuído e suportado pela Oracle Corporation. O servidor de bases de dados MySQL é muito rápido, escalável e fácil de utilizar. Em conjunto com o MySQL server, utilizou-se o MySQL Workbench que fornece ferramentas visuais para criar, executar e otimizar consultas SQL. Criou-se uma conexão local com as configurações mostradas na imagem abaixo.

Figura 4: Configurações da Conexão com o Banco de Dados no MySQL Workbench



Várias ferramentas do Workbench foram úteis durante o desenvolvimento do trabalho. Por exemplo, utilizamos o EERDiagram para fazer o modelo EER, Foward Engineer para auxílio na confecção das queries e o SQL Tab para fazer operações no banco de dados.

As queries de criação da base de dados e das tabelas, da inserção de dados nas tabelas e dos requisitos funcionais se encontram no repositório do projeto no GitHub no seguinte [link](#).