DETALHAMENTO DO PROJETO FÍSICO DE BANCO DE DADOS PARA SISTEMA DE ESTOQUE DE SUPERMERCADO DE BAIRRO

Sumário

[1 Introdução 3](#_heading=h.gjdgxs)

[2 Contextualização do problema 4](#_heading=h.30j0zll)

[3 Revisão do DER 5](#_heading=h.1fob9te)

[4 Construção das tabelas 8](#_heading=h.3znysh7)

[4.1 Normalização das tabelas 10](#_heading=h.2et92p0)

[4.1.1 Primeira Forma Normal – 1FN 10](#_heading=h.tyjcwt)

[4.1.2 Segunda Forma Normal – 2FN 12](#_heading=h.3dy6vkm)

[4.1.3 Terceira Forma Normal – 3FN 12](#_heading=h.1t3h5sf)

[5 Dicionário de dados 13](#_heading=h.4d34og8)

[6 Referências 14](#_heading=h.2s8eyo1)

[7 Anexos 15](#_heading=h.17dp8vu)

[7.1 ANEXO I 15](#_heading=h.3rdcrjn)

# 1 Introdução

Este documento tem como objetivo principal detalhar o projeto físico de um banco de dados para um sistema de estoque de supermercado de bairro. O projeto é baseado em um modelo conceitual desenvolvido durante a disciplina de Projeto Integrado I e envolve a definição de aspectos cruciais do banco de dados, tais como a criação das tabelas, sua normalização e um dicionário de dados com os tipos de dados, chaves primárias e estrangeiras, por exemplo. O foco é desenvolver um sistema eficiente que possa gerenciar de maneira eficaz o estoque de um supermercado de bairro.

No decorrer deste documento, vamos explorar cada aspecto do projeto do banco de dados em detalhes. Vamos definir as tabelas do banco de dados, especificar os tipos de dados para cada coluna, identificar as chaves primárias e estrangeiras e estabelecer as restrições de integridade necessárias. Além disso, vamos discutir a importância da normalização. Ao final, esperamos ter um projeto de banco de dados bem estruturado e otimizado para o sistema de estoque do supermercado.

# 2 Contextualização do problema

O Supermercado Bom de Preço, localizado no bairro Pirajá, na cidade de Juazeiro do Norte, é uma empresa familiar que tem prosperado ao longo de três décadas. Iniciado por Joaquim, o negócio cresceu de uma operação simples, gerenciada por ele e sua esposa Amália, para um estabelecimento próspero com quatro funcionários dedicados a várias funções, incluindo entregas, reposição de mercadorias e gerenciamento do frigorífico. Apesar da idade avançada, Joaquim ainda comanda o caixa do supermercado.

No entanto, o supermercado enfrenta desafios significativos em relação ao gerenciamento de estoque. Atualmente, todas as tarefas são realizadas manualmente, dependendo fortemente da atenção dos funcionários. O controle de estoque é feito através de uma planilha do Excel, que muitas vezes é esquecida de ser atualizada ao longo do dia, resultando em falta de mercadorias para reposição nas prateleiras. Além disso, a adição e remoção de novos itens muitas vezes causam problemas de dependência de dados. As entradas e saídas de produtos são raramente registradas, e o controle de entregas é praticamente inexistente, sendo feito com papel e caneta.

Eduardo, o neto de Joaquim e um entusiasta da tecnologia, sugeriu a implementação de um sistema de gerenciamento de estoque. Inicialmente, Joaquim estava relutante em mudar os processos que haviam servido ao negócio por 30 anos. No entanto, reconhecendo a necessidade de um sucessor para continuar seu projeto e a capacidade de Eduardo para assumir essa responsabilidade, Joaquim decidiu aceitar as mudanças propostas por seu neto.

Portanto, o Supermercado Bom de Preço requer um sistema de gerenciamento de estoque que armazene informações sobre produtos, fornecedores, clientes e funcionários. Além disso, o sistema deve registrar todas as movimentações de mercadorias do supermercado, incluindo solicitações de compra, recebimento de produtos e solicitações de venda. O sistema também deve fornecer o status do estoque, incluindo o estoque mínimo, máximo e a estoque atual.

# 3 Revisão do DER

Este documento de detalhamento do projeto físico de banco de dados para um sistema de estoque de supermercado de bairro começou com um modelo conceitual, materializado em um Diagrama de Entidade-Relacionamento (DER), documento inicial entregue em Projeto Integrado I. No entanto, durante o processo de transformação do modelo conceitual em um projeto físico de banco de dados nesta disciplina de Projeto Integrado II, foram identificadas oportunidades para melhorar o original. Essas melhorias foram incorporadas ao projeto, resultando em um DER aprimorado que atende melhor às necessidades do sistema de estoque do supermercado e visando também a aplicação de boas práticas tendo em vista o próximo capítulo de construção das tabelas.

O DER aprimorado é apresentado na Figura 1 do documento. Este capítulo detalha as melhorias feitas no diagrama original e explica como elas contribuem para a eficácia geral do sistema de estoque. Para fins de comparação e referência, a versão original do DER é fornecida no Anexo I deste documento. Isso permite que se veja claramente as mudanças feitas e entendam as razões por trás dessas alterações. A transparência e a justificativa dessas melhorias demonstram a evolução do projeto desde o seu conceito inicial até o projeto físico.

A diagram of a company

Description automatically generated

*Figura 1 - Novo Diagrama E-R*

A primeira mudança mais visível na revisão do diagrama antigo para o novo é que a entidade pessoa deixou de ter uma especialização/generalização. Isso foi feito pois, dessa forma, a modelagem lógica se mostrou mais eficiente e menos complexa. A eliminação da especialização/generalização simplifica o esquema do banco de dados, tornando-o mais fácil de entender e gerenciar. Além disso, isso pode melhorar o desempenho do banco de dados, pois menos junções são necessárias para consultar os dados.

A tabela estoque\_item do diagrama antigo foi incorporada à tabela Produtos do novo diagrama. Isso foi feito para eliminar a redundância de dados e simplificar o modelo de dados. Ao incorporar estoque\_item em Produtos, reduzimos o número de tabelas e garantimos que todas as informações sobre um produto estejam contidas em uma única tabela. Isso facilita a manutenção dos dados e melhora a eficiência das consultas ao banco de dados.

A relação entre as tabelas Fornecedores e Produtos foi otimizada de forma que a relação n para m foi eliminada e transformada em uma tabela própria de nome Produto\_Fornecedor. Isso é uma boa prática pois facilita o gerenciamento de relações muitos-para-muitos e melhora a integridade dos dados. Com uma tabela separada para gerenciar a relação, podemos garantir que cada combinação de produto e fornecedor seja única e possa ser gerenciada de forma independente. Isso também torna as consultas ao banco de dados mais eficientes, pois reduz a necessidade de junções complexas.

Em geral, os nomes das tabelas foram alterados para o plural, uma prática comum e recomendada em projeto de banco de dados. A razão para isso é que uma tabela é uma coleção de instâncias semelhantes, por exemplo, uma tabela de Clientes contém muitos clientes, uma tabela de Produtos contém muitos produtos, e assim por diante. Portanto, faz sentido nomear a tabela no plural, pois ela contém múltiplas instâncias de uma entidade. Essa convenção de nomenclatura ajuda a tornar o propósito da tabela imediatamente óbvio para qualquer pessoa que esteja lendo o esquema do banco de dados, melhorando assim a legibilidade e a manutenibilidade do banco de dados.

# 4 Construção das tabelas

A conversão do Diagrama de Entidade-Relacionamento (DER) para as tabelas do projeto físico do banco de dados é mais uma etapa fundamental no processo de projeto físico do banco de dados. Esta etapa envolve a tradução das entidades, atributos e relacionamentos representados no DER em tabelas, colunas, chaves primárias e chaves estrangeiras no banco de dados trazendo para um modelo lógico associado. Essa Modelagem Lógica também impactou no capítulo anterior, pois ao concebê-lo foram analisadas também entidades, suas cardinalidades e seus relacionamentos, de forma que foi visto na prática o que poderia funcionar melhor para a geração das tabelas tendo em vista o projeto físico.

Uma boa prática durante esta conversão é garantir que cada entidade e relacionamento no DER tenha uma representação correspondente no banco de dados. Isso ajuda a garantir que todas as informações necessárias sejam capturadas no banco de dados.

Com relação às chaves propriamente ditas, as chaves primárias foram projetadas, em termos de abordagem relacional, para serem o mais sucintas, isto é, devem obedecer ao princípio da minimalidade e mantém, por essa razão, a integridade de chave. Por outro lado, as chaves estrangeiras são o mecanismo para implementar de fato os relacionamentos entre as tabelas. Dessa forma, as chaves estrangeiras ajudam a manter a integridade referencial no banco de dados, ou seja, uma chave estrangeira em uma tabela aparecerá como chave primária em outra, garantindo que os relacionamentos entre as tabelas sejam mantidos corretamente.

Portanto é importante garantir que cada tabela no banco de dados tenha uma chave primária que identifique exclusivamente cada instância da tabela. Isso é especialmente importante para as tabelas que representam entidades no DER. Todo esse processo gerou as tabelas representadas na Figura 2.

A group of black and white text

Description automatically generated

*Figura 2 - Tabela convertidas após o Modelo Lógico*

## 4.1 Normalização das tabelas

Como parte final deste capítulo sobre as tabelas surge a tarefa importante garantir que o banco de dados esteja normalizado durante a conversão do DER para as tabelas do projeto físico. A normalização ajuda a minimizar a redundância de dados e a evitar problemas de atualização de dados. Isso pode envolver a divisão de tabelas em várias tabelas menores para eliminar a duplicação de dados. Também é importante equilibrar a normalização com o desempenho do banco de dados, pois a normalização excessiva pode levar a um grande número de junções de tabelas, afetando o desempenho das consultas ao banco de dados.

### 4.1.1 Primeira Forma Normal – 1FN

A Primeira Forma Normal (1FN) é uma etapa na normalização de um banco de dados. Ela estabelece os requisitos de banco de dados eficiente e sem redundâncias. Para uma tabela estar na 1FN, ela deve atender a quatro critério. Primeiro, deve existir uma chave primária (PK), que identifica unicamente cada registro na tabela. Segundo, a tabela deve possuir apenas valores atômicos, ou seja, cada campo deve conter um valor indivisível. Terceiro, a tabela não deve possuir atributos multivalorados, o que significa que cada campo deve conter um único valor, em vez de um conjunto de valores. Por fim, a tabela não deve possuir atributos compostos, ou seja, cada campo deve representar um único atributo, em vez de um grupo de atributos relacionados. Essas regras garantem que os dados sejam organizados de maneira lógica e consistente, facilitando a manutenção e a consulta dos dados. Com isso posto, pode-se analisar as tabelas em termos de sua primeira Forma Normal.

A tabela Endereços está na Primeira Forma Normal (1FN). Cada coluna contém apenas valores atômicos e a chave primária é ID\_Endereço.

A tabela Clientes também está na 1FN. Cada coluna contém apenas valores atômicos e a chave primária é ID\_Cliente. A coluna ID\_Endereço é uma chave estrangeira que referencia a tabela Endereços.

A tabela Funcionarios está na 1FN. Cada coluna contém apenas valores atômicos e a chave primária é ID\_Funcionario. A coluna ID\_Endereço é uma chave estrangeira que referencia a tabela Endereços.

A tabela Fornecedores está na 1FN. A chave primária é ID\_Fornecedor.

A tabela Produtos está na 1FN, pois cada coluna contém apenas valores atômicos e a chave primária é ID\_Produto.

A tabela Produto\_Fornecedor está na 1FN. As colunas ID\_Produto e ID\_Fornecedor são chaves primárias e estrangeiras que referenciam as tabelas Produtos e Fornecedores, respectivamente.

A tabela Pedido\_Cliente está na 1FN.

Cada coluna contém apenas valores atômicos e a chave primária é ID\_pedido\_cliente. As colunas ID\_Cliente, ID\_Produto e ID\_Funcionario são chaves estrangeiras que referenciam as tabelas Clientes, Produtos e Funcionarios, respectivamente.

A tabela Solicitacoes\_Compras está na 1FN. Cada coluna contém apenas valores atômicos e a chave primária é ID\_solicitacao\_compra. As colunas ID\_Funcionario, ID\_Produto e ID\_Fornecedor são chaves estrangeiras que referenciam as tabelas Funcionarios, Produtos e Fornecedores, respectivamente.

A tabela Produtos\_Recebidos está na 1FN. Cada coluna contém apenas valores atômicos, a chave primária é ID\_Produtos\_Recebidos. A coluna ID\_Solicitação\_Compra é uma chave estrangeira que referencia a tabela Solicitacoes\_Compras.

Em conclusão, todas as tabelas estão na Primeira Forma Normal (1FN), pois cada coluna em cada tabela contém apenas valores atômicos, cada coluna tem um nome único dentro de sua tabela e as linhas são únicas.

### 4.1.2 Segunda Forma Normal – 2FN

A partir de agora, por não se tratar de um problema puramente de engenharia reversa, apesar de termos aplicado alguns de seus conceitos durante o retrabalho dos modelos anteriores, a segunda forma normal foi contemplada em todas as tabelas, pois possuem apenas uma coluna como chave primária. Portanto, as tabelas não contêm dependências parciais, ou seja, as colunas dependem de uma única chave primário não composta.

### 4.1.3 Terceira Forma Normal – 3FN

Na passagem das tabelas da 2FN para a 3FN o objetivo é eliminar as dependências transitivas, isto é, aquelas em que uma coluna não chave primária depende de outra ou uma combinação de colunas não chave.

Ao analisar as tabelas da 2FN foi constatado que não há tais dependências, estando as tabelas, portanto, em sua terceira forma normal, 3FN.

# 5 Dicionário de dados

# 6 Referências

HEUSER, Carlos Alberto. **Projeto de banco de dados**. 6. ed. Porto Alegre: Bookman, 2009. 281 p. Recurso eletrônico. Disponível em: https://app.minhabiblioteca.com.br. Acesso em: 11 jun. 2024.

REIS, Fábio dos. **Curso de modelagem de dados**. Youtube, 11 jul. 2017. Disponível em: <https://youtube.com/playlist?list=PLucm8g\_ezqNoNHU8tjVeHmRGBFnjDIlxD&si=fPHWmLwOZKbti3k6>. Acesso em 1 jun. 2024.

# 7 Anexos

## 7.1 ANEXO I

A diagram of a computer program

Description automatically generated