

DADOS E MODERNIZAÇÃO:

UMA ABORDAGEM PRÁTICA NA IMPLEMENTAÇÃO DE

BANCO DE DADOS

Francisco Gabriel de Araújo Chagas¹

Gerson Bhrener Silva Ribeiro de Sá¹

Gianmarco Salvador Daitx¹

Carcius Clayton Andrade Santos Jacinto²

1. INTRODUÇÃO

Quando observamos o passado, notamos que durante muito tempo o homem buscou meios para registrar as “variáveis” do seu cotidiano; um exemplo muito conhecido são as pinturas rupestres do período da pré-história da humanidade. Mais à frente, houve muitas outras formas de registrar os dados: papiros, pergaminhos, livros, pedras, tatuagens etc. O fato é, se homem não tivesse encontrado uma forma de gravar seus dados para a sua posteridade, jamais teríamos nos estabelecido como civilização. Neste seminário exploraremos mais a fundo os bancos de dados relacionais e suas principais características, assim como, implementaremos uma solução prática simples de um banco de dados fictício, passando por todas as etapas do projeto, desde a modelagem conceitual até a solução implementada em um dos mais famosos SGBDs atualmente: o PostgreSQL.

¹ Francisco Gabriel de Araújo Chagas

¹ Gerson Bhrener Silva Ribeiro de Sá

¹ Gianmarco Salvador Daitx

² Carcius Clayton Andrade Santos Jacinto

2. FUNDAMENTAÇÃO TEÓRICA

2.1. Um breve histórico dos dados e bancos de dados relacionais

Mas qual o significado de dados para a vida do homem? Segundo Machado e Abreu (1996), dado é apenas uma representação, ou seja, concerne um registro de uma informação. Além do mais, o mesmo pode ser arquivado em elementos físicos por meio de um papel, um HD (Hard disk) de computador etc. Os mesmos autores ainda afirmam que os dados podem originar uma série de processos que influenciam na realidade observada; por exemplo: salvar a vida de uma paciente, tocar um alarme etc.

Os dados isolados não compreendem um fato relevante, porém “[...] o DADO deve registrar apenas os aspectos realmente relevantes da INFORMAÇÃO [...]” (MACHADO; ABREU, 1996, p.1). Com isso, diferentemente dos dados: “A INFORMAÇÃO acrescenta algo ao conhecimento da realidade a ser analisada. Por exemplo, a dosagem que um paciente precisa receber de um determinado remédio, é uma INFORMAÇÃO”. (MACHADO; ABREU, 1996, p.1).

É fato que, apesar de toda a sociedade se beneficiar dos dados e informações no dia a dia, são os grandes empreendimentos que mais tem esmero por eles, uma vez que estes, são “pontas de lança” para gerar conhecimentos valiosos para os seus negócios, de modo a propiciar que estes negócios sejam alcançados, ou, num contexto mais otimista, superem as suas expectativas. Segundo o site Dicas de Programação (2013):

Antigamente as empresas armazenavam dados em fichas de papel que eram organizadas em arquivos físicos através de pastas. Extrair informações e manter esses arquivos organizado era uma tarefa muito custosa. Além disso o acesso à informação dependia da localização geográfica dos arquivos. Enfim esses arquivos físicos evoluíram para arquivos digitais.

“Na década de 60 a empresa IBM investiu fortemente em pesquisas para solucionar estes problemas dos bancos de dados digitais primitivos. Vários modelos de bancos de dados surgiram nesta época, dentre eles os modelos hierárquico[sic] e rede.” (DICAS DE PROGRAMACAO, 2013).

Na década de 70 o pesquisador da IBM Edgar Frank Codd propôs “[...] o modelo de dados relacional, que se tornou um marco em como pensar em banco de dados. Ele desconectou a estrutura lógica do banco de dados do método de armazenamento físico. Este sistema se tornou padrão desde então.” (DEVMEDIA, 2006).

Nos anos seguintes, surgiram muitas ferramentas para a modelagem e implementação dos bancos de dados relacionais propostos por Codd: os famosos SGBDs, ou Sistemas Gerenciadores de Banco de Dados. Nesse período, mais precisamente na década de 90, houve uma grande

movimentação por conta da Internet em função da WWW (World Wide Web); a grande teia mundial que foi um marco na implementação dos bancos de dados relacionais pelos mais diversos sistemas espalhados pelo mundo.

Atualmente existem vários modelos de bancos de dados tais como: orientado a objetos, orientado a documentos, etc. “Mas o mais comum ainda é o banco de dados relacional. A decisão entre qual modelo de banco de dados utilizar baseia-se no tipo de dados que você pretende armazenar.” (DICAS DE PROGRAMACAO, 2013).

2.2 Modelagem de banco de dados

2.2.1 Modelos de bancos de dados na perspectiva entidade-relacionamento

O MER ou Modelo Entidade-Relacionameto representa de forma gráfica os objetos das estruturas do mundo externo, ou seja, o real. Assim sendo, o modelo apresenta a forma estrutural em sua forma conceitual e sua lógica como um todo. Desta forma, segundo Machado e Abreu (2004), a Modelagem conceitual de Dados com a técnica de Entidades e Relacionamentos, obteremos resultados e esquemas puramente conceituais sobre a essência de um sistema, ou melhor, sobre o negócio para o qual estamos desenvolvendo um projeto

Existem basicamente três etapas em um projeto de bancos de dados relacional, que são: conceitual; lógico; físico. “Um modelo de banco de dados é uma descrição dos tipos de informações que estão armazenadas em um banco de dados.” (HEUSER, 2009, p.24). Neste contexto, para a criação de um modelo de banco de dados se faz necessária a utilização de uma linguagem de modelagem de dados que segundo Heuser (2009), podem ser classificadas como *textuais* ou *linguagens gráficas*.

Antes de abordarmos os tipos de modelos em um projeto de banco de dados, é importante definirmos os principais elementos que darão suporte para a sua construção. “Os modelos de dados conceituais utilizam conceitos como entidades, atributos e relacionamentos.” (ELMASRI; NAVATHE, 2011, p.20). E como estes elementos se constituíram ao longo do nosso projeto.

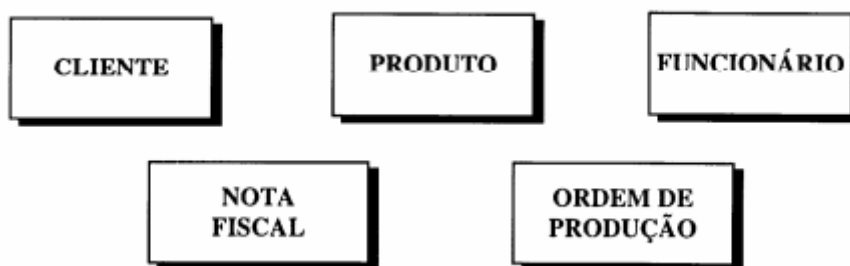
2.2.2 Entidade

As entidades são, para Machado e Abreu (2004), os objetos que existem no mundo real que dispõem de uma identificação distinta, mas que tem um significado próprio. Elmasri e Navathe (2011)

reiteram Machado e Abreu ao afirmarem que uma entidade representa um objeto ou conceito do mundo real, ou seja, pode representar um funcionário por exemplo.

As entidades são divididas em: entidade tipo; entidade fraca ou dependente; entidade associativa; entidade agregada; e por fim, entidade subordinada. Para os fins deste trabalho que se constitui em uma abordagem mais prática, não entraremos a fundo nestas diferenças.

FIGURA 1 - EXEMPLOS DE ENTIDADES



Fonte: ABREU; MACHADO (2004, p. 35)

Na imagem acima é possível identificar alguns exemplos de entidades. As entidades devem ser representadas por substantivos no singular. A entidade **CLIENTE**, obviamente, representará todos os clientes de um negócio. A entidade **PRODUTO** se refere aos produtos que poderão ser entregues aos clientes ou ainda poderão ser comprados da entidade **FORNECEDOR** que não está presente na imagem. A entidade **FUNCIONÁRIO** se refere aos funcionários físicos e suas informações; essa entidade será muito importante para gerenciar a presença dos funcionários diariamente. A **NOTA FISCAL** representará as informações que serão impressas futuramente. Por fim, a **ORDEM DE PRODUÇÃO** será importante para indicar o que deve ser feito. Estas entidades sozinhas não têm função nenhuma, mas a ideia do modelo relacional percorrido nesse trabalho é justamente explicar que: é a relação entre estas que fazem o sistema dinâmico e inteligente.

2.2.3 Atributo

“Todo objeto para ser uma entidade possui propriedades que são descritas por atributos e valores.” (ABREU; MACHADO, 2004, p. 35). “Um atributo representa alguma propriedade de interesse que descreve melhor uma entidade, como o nome ou o salário do funcionário.” (ELMASRI; NAVATHE, 2011, p. 20). Além disso, pode ser visto como, “[...] uma característica ou qualidade de uma entidade que tem valor para o negócio, que deve ser observada pelo usuário.” (SIQUEIRA, ?).

Os atributos, assim como as entidades, são subdivididos em subtipos. Entender os seus conceitos é importante para evitar as “armadilhas” que impedem uma boa implementação do projeto e pior ainda ocasionam muitas modificações em etapas anteriores.

Os atributos entidade-relacionamento se dividem em:

- Atributos Simples ou Atômicos: São os atributos que não são divisíveis. Como por exemplo, o atributo CPF, este não pode ser dividido em mais atributos.
- Atributos Compostos: Podem ser divididos em várias partes com significados independentes. O atributo Endereço de um funcionário, por exemplo, pode ser dividido em Rua, Número e Bairro.
- Atributos Monovalorados: São aqueles que possuem apenas um valor para aquela entidade. O atributo número da casa, por exemplo, só pode receber um valor.
- Atributos multivalorados: podem assumir diversos valores. Como por exemplo, o atributo Telefone de um funcionário, que pode assumir diversos valores.
- Atributos Derivados: Em alguns casos, dois ou mais atributos estão relacionados, por exemplo, a idade e data de nascimento de um funcionário. Para um determinado funcionário, podemos determinar a sua idade por meio da data de nascimento e da data atual. Atributos como a idade são chamados de atributos derivados.
- Atributo-chave: Tipo de atributo cujos valores são distintos para cada entidade em um conjunto de entidades. O RG de um funcionário, por exemplo, constitui um atributo-chave para o tipo de entidades FUNCIONÁRIO. Algumas vezes, um conjunto de atributos pode formar uma chave. Mais do que isso, alguns tipos de entidades podem ter mais que um atributo-chave. (PORTAL EDUCAÇÃO, 2015).

2.2.4 Relacionamentos

Os relacionamentos no modelo entidade-relacionamento são como as entidades estão associadas umas com as outras. Essa etapa é crucial em um projeto de banco de dados, pois é nela que vamos configurar inúmeras situações que serão importantes no momento da implementação do banco de dados na camada lógica e física do projeto.

“O entendimento sobre o que são efetivamente relacionamentos e a capacidade de enxergar estes objetos, como participantes do mundo real, são fatores primordiais para que se venha a efetuar trabalhos de modelagem com compreensão do que está sendo realizando [sic].” (ABREU; MACHADO, 2004, p. 45).

Os relacionamentos também apresentam o “grau”. “O grau dos relacionamentos corresponde ao número de entidades envolvidas na mesma relação.” (BLRDATA, 2016). O grau de um relacionamento pode ser: **Binário**, onde duas ou mais entidades se relacionam; **Ternário**, onde três entidades participam do relacionamento; **N-ário**, onde quatro ou mais entidades se relacionam.

2.2.5 Cardinalidade

“Para fins de projeto de banco de dados, uma propriedade importante de um relacionamento é a de quantas ocorrências de uma entidade podem estar associadas a uma determinada ocorrência através do relacionamento.” (HEUSER, 2009, p. 39). “Esta propriedade é chamada de cardinalidade de uma entidade em um relacionamento. Há duas cardinalidades a considerar: a cardinalidade máxima e a cardinalidade mínima.” (HEUSER, 2009, p. 39).

“A cardinalidade mínima é responsável por orientar a obrigatoriedade (opcionalidade) do relacionamento. Já a cardinalidade máxima é responsável por definir a quantidade máxima de vezes que um elemento pode estar associado no relacionamento.” (BLRDATA, 2016).

O relacionamento com cardinalidade de um-para-um ou 1:1, representa a relação entre duas entidades onde, “[...] cada elemento de uma entidade relaciona-se com um e somente um elemento de outra entidade.” (ABREU; MACHADO, 2004, p. 54).

O próximo relacionamento, quanto ao grau, é o de um-para-muitos ou 1:N ou N:1. “Um elemento da entidade 1 relaciona-se com muitos elementos da entidade 2, mas cada elemento da entidade 2 somente pode estar relacionado a um elemento da entidade 1.” (ABREU; MACHADO, 2004, p. 56). E vice-versa, isso depende da ordem das entidades no modelo, a mesma leitura feita da esquerda para a direita, como dizem os autores acima, pode ser feita da direita para a esquerda dependendo da situação.

Por fim, o relacionamento de grau muitos-para-muitos ou N:N. Acontece quando vários elementos de uma entidade X se relacionam com vários elementos da entidade Y. Vale ressaltar que desse grau de relacionamento, origina-se uma entidade “intermediária” ou entidade associativa, a qual será abordada com mais detalhes durante a explanação da prática deste trabalho.

2.2.6 Modelo Conceitual

O modelo conceitual dá início a um projeto de banco de dados, é nele onde os desenvolvedores irão se atentar às regras de negócio e as necessidades particulares dos clientes. Além disso, essa etapa definirá a próxima, mas também não significa que esta não possa ser revisada e melhorada futuramente em detrimento de novos requisitos ou a revisão destes.

“A modelagem conceitual basea-se [sic] no mais alto nível e deve ser usada para envolver o cliente, pois o foco aqui é discutir os aspectos do negócio do cliente e não da tecnologia.” (UTILIDADE PUBLICA, 2021). E ainda que, “Os exemplos de modelagem de dados vistos pelo modelo conceitual são mais fáceis de compreender, já que não há limitações ou aplicação de tecnologia específica.” (UTILIDADE PUBLICA, 2021)

Segundo Machado e Abreu (2004), o Modelo Conceitual tem como objetivo:

[...] descrever as informações contidas em uma realidade, as quais irão estar armazenadas em um banco de dados. É uma descrição em alto nível (macrodefinição), mas que tem a preocupação de captar e retratar toda a realidade de uma organização, setor, repartição, departamento, etc.

Elmasri e Navathe (2011), trazem um contraponto ao modelo ao afirmarem que muitas especializações e subclasses podem ser definidas de modo a tornar o modelo preciso. Todavia, a desvantagem é que essa abordagem torna o projeto muito confuso. Para tanto, os autores indicam que se faz importante representar as subclasses necessárias de modo a evitar o acúmulo e aglomeração deste modelo.

2.2.7 Modelo Lógico

Em sequência, temos o modelo lógico, que corresponde a segunda etapa do projeto de banco de dados na visão entidade relacionamento.

“Um modelo lógico é uma descrição de um banco de dados no nível de abstração visto pelo usuário do SGBD. Assim, o modelo lógico é dependente do tipo de SGBD que está sendo usado.”(HEUSER, 2009, p. 26). E autor procede:

O modelo lógico descreve a estrutura do banco de dados, conforme vista pelo usuário do SGBD. Detalhes de armazenamento interno de informações, que não têm influência sobre a programação de aplicações no SGBD, mas podem afetar o desempenho da [sic] aplicações (por exemplo, as estruturas de arquivos usadas no acesso às informações) não fazem parte do modelo lógico. (HEUSER, 2009, p. 27)

2.2.8 Modelo Físico

Finalmente, o modelo físico marca a “conclusão” de um projeto de banco de dados. Segundo o site Utilidade Pública (2021), o modelo físico marca o processo de modelagem física do nosso banco de dados. Dessa forma, devemos levar em conta alguns aspectos, como as limitações do SGBD escolhido.

A modelagem física de dados é o terceiro de três estágios sequenciais na modelagem de dados. Designers de banco de dados produzem modelos físicos de dados elaborando os modelos criados nas etapas de modelagem conceitual e lógica de dados. Os modelos criados nesta fase permitem a desnormalização gerenciada e levam em consideração a tecnologia alvo para implementação. Eles são completos o suficiente para representar o design do banco de dados como implementado, ou como pretendido para ser implementado. (ERWIN, 2021).

A descrição deste modelo é feita por meio de códigos SQL (Structured Query Language) e, como exposto acima pelos autores, este código ditará ao SGBD (MySQL, SQL Server, MariaDB, PostgreSQL etc) escolhido o que ele deve criar, modificar ou excluir. Dependendo do projeto de banco de dados e da qualidade dele, essa etapa pode ser até mesmo automatizada, mas para tanto, é necessária a justa adesão às etapas anteriores. A figura abaixo descreve um trecho de código da codificação da tabela turma em um banco de dados MySQL:

2.2.9 Normalização

A Microsoft (2021) conceitua a normalização como o processo de organização de dados em um banco de dados. Estão inclusos: a criação de tabelas e o estabelecimento de relações entre essas tabelas de acordo com as normas projetadas para proteger os dados e tornar o banco de dados mais flexível, eliminando a redundância e a dependência inconsistente.

Segundo a Microsoft (2021), existem algumas regras para se fazer a normalização de um banco de dados. Cada regra é denominada de “forma normal”. Se a primeira regra foi observada no banco de dados, dizemos que ele está na “primeira forma normal”. Se o encontramos três regras nele, dizemos que ele está na “terceira forma normal”. Apesar de serem possíveis aplicar outras camadas de regras, a terceira forma normal é considerada a camada mais alta para a maioria das aplicações. Um ponto a se ressaltar, é que um banco de dados não poderá estar na segunda forma normal, sem antes estar de acordo com a primeira, ou na terceira, sem antes ter passado pela primeira e segunda. Em outras palavras, a normalização é sequencial e não deve saltar etapas. A seguir veremos como são definidas as três principais formas normais.

A primeira forma normal ou 1FN consiste em: eliminar grupos repetidos em tabelas, como vários contatos de telefone na mesma tabela por exemplo; criar uma tabela separada para cada conjunto de dados relacionados, ou seja, uma tabela para telefone, endereço etc; e identificar cada conjunto de dados relacionados com uma chave primária.

A segunda forma normal ou 2FN consiste em: criar tabelas separadas para conjuntos de valores que se aplicam vários registros, como a tabela telefone citada na 1FN; relacionar essas tabelas com uma chave estrangeira, ou seja, conectá-las através das chaves primárias e estrangeiras.

Por fim, a terceira forma normal ou 3FN, que de modo direto, consistem em eliminar os campos que não dependem do atributo chave.

Por exemplo, em uma tabela de Recrutamento de Funcionários, o nome da universidade e o endereço de um candidato podem ser incluídos. Mas você precisa de uma lista completa de universidades para envios de mensagens em grupo. Se as informações da universidade são armazenadas na tabela Candidatos, não há como listar universidades sem candidatos atuais. Crie uma tabela de Universidades separada e vincule-a à tabela Candidatos com uma chave de código da universidade. (MICROSOFT, 2021)

2.3 Criação e Automação do Projeto de Banco de Dados com o BrModelo

O BrModelo surgiu de um “projeto de conclusão de curso com o objetivo de criar uma ferramenta para o modelo conceitual é o foco desta aplicação ao contrário das principais ferramentas disponíveis no mercado.” (ROMANO, 2011, p. 44).

Suas vantagens em relação às ferramentas avaliadas, em síntese, são:

- Permitir alterações estruturais no modelo diante de novas decisões do analista;
- Conversão de atributo em entidade;
- Conversão de relacionamento em entidade associativa;
- Conversão de especialização de restrita para opcional ou vice-versa.
- Uma outra importante funcionalidade é a capacidade de exibir seus modelos lógicos (salvos em XML) [...];
- Não há necessidade de instalação, pode rodar direto do pen-drive. (ROMANO, 2011, p. 44).

De longe a principal funcionalidade do BrModelo, além da facilidade na criação dos modelos (conceitual, físico e lógico), é a sua capacidade de converter dinamicamente um modelo no outro. É óbvio que para tanto, é necessário seguir etapas de forma sequencial de acordo com o que normalmente acontece nas outras ferramentas. A partir do modelo conceitual é possível converter dinamicamente para o modelo lógico sem precisar reescrever tudo novamente; e do modelo lógico é possível gerar todo o código necessário para ser inserido no SGBD, com as características e restrições definidas nas etapas anteriores.

É importante frisar que o código SQL gerado pode não ser compatível com todos os SGBDs, todavia, sua geração automática a partir de um modelo prévio, já suplanta uma grande porcentagem do trabalho manual para escrevê-lo linha a linha; restando para os desenvolvedores pequenas modificações.

2.4 Sistemas Gerenciadores de Banco de Dados Relacionais (SGBD)

“Um sistema gerenciador de banco de dados (SGBD — Database Management System) é uma coleção de programas que permite aos usuários criar e manter um banco de dados.” (ELMASRI; NAVATHE, 2011, p.3). “O SGBD é um sistema de software de uso geral que facilita o processo de

definição, construção, manipulação e compartilhamento de bancos de dados entre diversos usuários e aplicações.” (ELMASRI; NAVATHE, 2011, p.4).

Atualmente, existem inúmeros tipos de SGBDs espalhados pela Internet, uns mais robustos, outros mais simples. Os principais do mercado são: MySQL; ORACLE DB; SQLServer; MariaDB; PostgreSQL. A escolha de um ou mais SGBDs não depende de uma receita pronta, mas cabe aos desenvolvedores escolherem a melhor solução, ou, as melhores soluções para o seu projeto.

3 RESULTADOS E DISCUSSÕES

Em uma cultura moderna onde empresas estão se tornando cada vez mais “data-driven”, ou seja, com seus planejamentos e decisões são orientados a dados, cuidar deste ativo digital se tornou imprescindível para a condução às melhores estratégias e recursos resultando consequentemente em resultados embasados não apenas na opinião do gestores e suas intuições mas em informações relevantes e concretas do histórico do empreendimento capacitando os recursos humanos a ter uma visão mais ampla, objetiva e holística do negócio.

O desafio é gigantesco. A geração de dados, com advento e massificação dos sistemas computacionais e alto avanço da tecnologia em seus variados aspectos, está aumentando cada vez mais a níveis jamais vistos, demonstrado com clareza nos recentes fenômenos da big data e o desafio se mostra claramente em como lidar, tratar e extrair insights num mundo tão diverso, as vezes confuso e de proporções. De acordo com o Instituto Gartner, umas das maiores consultorias em tecnologia da informação do mundo, estima-se que a cada 6 minutos é gerado no universo virtual 9,1 mil terabytes de dados e é sabido que as empresas não absorveram o volume de dados de forma proporcional ao seu crescimento e neste caso são necessárias novas técnicas e ferramentas que as ajudem a enfrentar este gigante do mundo moderno.

E neste trabalho específico visamos mostrar uma destas técnicas, de forma direta, objetiva um dos primeiros passos rumo à direção já citada por meio da implementação de um sistema gerenciador de banco de dados.

Neste estudo em específico, utilizamos de uma empresa fictícia de ramo diverso que controla todas as suas movimentações de estoque, vendas, clientes e recursos pessoais por meio de planilhas demonstrando claramente sua falta atualização frente ao mercado tecnológico. As planilhas possuem muitas vantagens seja para organização de poucos dados ou de dados individuais de cada funcionário, cálculos rápidos e processos de análise mais simplificado, entretanto gerenciar toda uma organização por meio delas é assumir um risco incalculável frente à sensibilidade do conteúdo gerado por uma empresa em seus processos diários. Além disso, planilhas possuem pouca estabilidade permanecendo no desktop de quem utiliza ou em processos de colaboração utilizando ferramentas como o Google Sheets. A segurança é prejudicada quanto a sua integridade já que planilhas podem se corromper e causar perdas irreparáveis ao negócio em si bem como a limitação do que se pode extrair já que estes mecanismos são fragmentados seja por pessoal, departamental ou setorial. Desta feita, a implementação de um SGBD é primordial para a melhoria na condução, acesso, tratamento e gravação das informações permitindo um ambiente mais sólido na gestão de dados.

No processo de instalação de um SGBD é primordial o estudo das necessidades da empresa já que soluções são muitas e cada uma delas pode atender a certas demandas específicas de cada organização. Neste estudo foi utilizado o PostgreSQL, em instalação no desktop Windows para fins de simulação, e escolhido por sua leveza, rapidez e facilidade de administração.

Após a instalação do SGBD um dos passos mais importantes na implementação de um banco de dados é justamente fazer a modelagem destes com o objetivo de identificarmos entidades e seus atributos assim como a relação existente entre si com o objetivo de analisar e explorar os aspectos comportamentais dos dados à medida que são gerados pela organização. Outra explicação da modelagem é criar uma abstração da realidade que seja capaz de registrar este comportamento. Aqui passamos por três fases: a modelagem conceitual, a modelagem lógica e por fim, a modelagem física que se diferem pelo nível de detalhamento. Durante a condução das fases conceitual e lógica utilizaremos o tão conhecido MER ou Modelo Entidade-Relacionamento que é uma representação gráfica dos objetos do mundo real sendo de grande ajuda em sua abstração permitindo uma comunicação efetiva junto a desenvolvedores e analistas. Para este fim foi utilizado o software bRModelo.

A começar pela fase conceitual onde transcrevemos para o modelo uma descrição de alto nível da realidade dos dados damos início a identificação e representação gráfica das entidades (objetos físicos ou abstratos do mundo real de existência própria) e do como se comunicam por meio das relações. No processo de transição é de suma importância a análise e identificação das entidades corretas que correspondem o objeto de estudo em si. Na planilha de gerenciamento de vendas podemos identificar entidades bem características deste universo: gerente, cliente (físico e jurídico aqui chamadas de especializações), pedido, produto e aquisição. Note que “aquisição” é um conceito abstrato, porém como a partir dela se gera o pedido a chamamos de entidade associativa. Os relacionamentos como já explicados fazem a conexão das entidades: gerente cuida/gerencia/trata com um cliente, cliente adquire produto, a aquisição gera pedido.

A transição do modelo conceitual para o modelo lógico é marcada por um conceito muito importante chamado de cardinalidade. Ela é definida pela quantidade de interações que as entidades têm sobre si. Uma ilustração prática do nosso estudo em particular: um gerente pode tratar com nenhum, um ou mais assim como um cliente só pode ser gerenciado por um e apenas um gerente. Essa dinâmica entre as entidades se constitui como o cerne da cardinalidade.

Entramos assim na fase lógica onde representamos as estruturas de armazenamento de dados, comumente chamada de forma tabular já visando a escolha do SGBD que neste paper é demonstrado pela abordagem Relacional, e detalhamos mais o modelo citando também os atributos relativos a cada

entidade. Em nosso estudo a entidade cliente possui: nome, endereço, documentação e seu tipo. Gerente possui nome. Produto: nome, categoria, tipo de embalagem etc.

Por fim a etapa física onde o intuito visa a aplicação no SGBD e tem o caráter mais específico das fases. O tipo dos dados, o tamanho de memória, restrições e relacionamentos por meio das chaves são definidos nesta fase. Estes conceitos variam muito dependendo do gerenciador a ser escolhido pois todos possuem formas específicas de definição. Nesta fase que se apresenta a linguagem SQL para a manipulação dos dados no SGBD. No nosso estudo, a transição dos dados foi feita no próprio Excel por meio de técnicas de limpeza com uso de fórmulas e extraída e transformada em código SQL por meio da ferramenta SQLizer disponível neste link: <https://sqlizer.io/>. A estruturação do modelo físico e sua implementação SQL ficou a cargo do brModelo.

Cabe ressaltar ainda o conceito de normalização que visa diminuir as ocorrências de duplicidade, consistência e integridade dos dados. Em nosso estudo vemos que determinados valores como o preço total do pedido dependem da quantidade que foi pedido assim como seu preço unitário. Valores já calculados não possuem a necessidade de estar em um banco de dados já que por meio da linguagem SQL e suas funções de agregação é possível realizar estes mesmos cálculos retornando o resultado por meio do comando SELECT. Este é um dos exemplos onde a normalização é de suma importância.

Após todos estes passos segue a instalação e implementação física do sistema de banco de dados. Como já citado é interessante analisar as demandas e necessidades da organização para uma escolha assertiva. Em nosso estudo foi utilizado o PostgreSQL em modelagem relacional, mas de acordo com as necessidade pode-se e deve levar em consideração outras abordagens como por exemplo bancos NoSQL. Entrevista, pesquisas, diálogos entre departamentos são de grande importância. O local de implementação também deve ser observado: seja localmente por meio de instalação nos servidores físicos, na infraestrutura local da empresa ou como serviço em nuvem, por meio de máquinas virtuais para uma maior distribuição. Além disso, a segurança e integridade dos dados deverão ser observados com o controle de acesso que irá definir quem ou quais departamentos podem acessar o banco e realizar modificações e quais dados deverão ser disponibilizados de acordo com as regras e políticas de privacidade instauradas assim como a definição de backup periódicos.

Os ganhos com a implementação de banco de dados são muitos. Melhoria do relacionamento e produtividade na empresa com o acesso concorrente, aumento da eficiência e desempenho pela velocidade de acesso e consistência dos dados além de conseguir gerar insight melhores com uma visão mais abrangente definindo estratégias com melhor embasamento dada amplitude de informações geradas pelos dados. Como serviço se destacam a redução de custos dada a não necessidade de investimento em infraestrutura local, a escalabilidade que se adapta a demanda

específica da organização como grandes benefícios da implementação em nuvem impulsionando de forma impressionante a elasticidade da integração com outros aplicativos.

4 REFERÊNCIAS

ALVES, Gustavo Furtado de Oliveira. A história dos bancos de dados. **Dicas de Programação**, 1 abr. 2013. Disponível em: <https://dicasdeprogramacao.com.br/a-historia-dos-bancos-de-dados/>. Acesso em: 9 out. 2021.

DESCRIÇÃO das noções básicas de normalização do banco de dados. **Microsoft: Docs**, [s. l.], 12 set.2021. Disponível em: <https://docs.microsoft.com/pt-br/office/troubleshoot/access/database-normalization-description>. Acesso em: 13 nov. 2021.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de banco de dados**. Tradução: Daniel Vieira. 6. ed. rev. São Paulo: Pearson, 2011. 788 p. ISBN 978-85-4301-381-7.

HEUSER, Carlos Alberto. **Projeto de banco de dados**. 6. ed. Porto Alegre: Bookman, 2009. 282 p. ISBN 978-85-7780-452-8.

LOPES, Bergson. Modelo Conceitual de Dados: Aprenda a utilizar os principais mecanismos de abstração. **BLRDATA: BLOG DA BLR DATA**, 19 mar. 2016. Disponível em: <https://www.blrdata.com.br/single-post/2016/03/19/modelo-conceitual-de-dados-aprenda-a-utilizar-os-principais-mecanismos-de-abstra%C3%A7%C3%A3o>. Acesso em: 13 nov. 2021.

MACHADO, Felipe Nery Rodrigues; ABREU, Maurício Pereira. **Projeto de Banco de Dados: Uma Visão Prática**. 11. ed. São Paulo: Érica, 2004. 297 p. ISBN 85.7194.312-5.

MODELAGEM de dados: modelo conceitual, modelo lógico e físico. **Utilidade Pública**, 2021. Disponível em: <https://www.luis.blog.br/modelagem-de-dados-modelo-conceitual-modelo-logico-e-fisico.html>. Acesso em: 13 nov. 2021.

MODELAGEM física de dados. Erwin, 2021. Disponível em: <https://www.erwin.com/br-pt/solutions/data-modeling/physical.aspx>. Acesso em: 13 nov. 2021.

REZENDE, Ricardo. A História dos Banco de Dados. **Devmedia**, 2006. Disponível em: <https://www.devmedia.com.br/a-historia-dos-banco-de-dados/1678>. Acesso em: 9 out. 2021.

ROMANO, Simone Maria Viana. **Banco de dados I**. São Paulo: Fatec, 2011. v. 1.

TIPOS de Atributos. **Portal Educação**, 24 ago. 2015. Disponível em: <https://siteantigo.portaleducacao.com.br/conteudo/artigos/informatica/tipos-de-atributos/66721>. Acesso em: 13 nov. 2021.