



Modelagem de Dados

Sumário

Apresentação	4
Módulo 1	6
Introdução ao banco de dados	6
Introdução ao banco de dados	7
Banco de dados	7
Minimundo e abstração	10
Sistema de Gerenciamento de Banco de Dados (SGBD)	14
Modelo conceitual	18
Modelo físico	22
Vantagens de utilizar um banco de dados	23
Tabelas, campos e registros	29
Domínio	31
Chave primária	33
Chave estrangeira	35
Normalização	37

Sumário

Módulo 2	43
Relacionamentos	43
Relacionamentos	44
Modelo Entidade Relacionamento (MER)	44
Elementos do MER	46
Condicionalidade de um relacionamento e autorrelacionamento	55
Módulo 3	58
Integridade de um banco de dados e outros conceitos	58
Integridade de um banco de dados e outros conceitos	59
12 regras de Codd para bancos relacionais	59
Restrições de integridade	61
Fechamento	66
Referências	67

Apresentação

Olá!

Bem-vindo(a) ao curso **Modelagem de Dados!**

O objetivo aqui é entender que, com a evolução tecnológica dos últimos anos, tornou-se essencial a administração e estocagem de dados. Assim, a informação, considerada um dos bens mais preciosos para as organizações, passou a ter um peso essencial nesse cenário, visto que auxilia nas tomadas de decisões. Por outro lado, para uma pessoa física, suas informações pessoais (acessos a sistemas, documentos, entre outros) são cruciais para a rotina diária.

Visando essa realidade, o curso trará mais detalhadamente a explicação de como realizar a modelagem de dados e quais técnicas estão envolvidas, incluindo, ainda, temas como a teoria relacional, a normalização, o autorelacionamento, a cardinalidade, entre outros conceitos.

Desejamos a você um bom aprendizado!



Vídeo

Confira o [vídeo](#) de apresentação do curso.

Perdeu algum detalhe? Confira o que foi abordado no vídeo.

Olá, bem-vindo(a) ao curso de Modelagem de Dados.

Neste curso, você verá uma introdução a respeito do banco de dados, de modo a conhecer ou recordar os conceitos relacionais e suas vantagens. Mais para frente, falaremos sobre minimundo e abstração, sistema de gerenciamento de banco de dados, além dos modelos conceitual, lógico e físico.

Outro ponto dos estudos será quanto aos bancos de dados relacionais e suas particularidades, em que você poderá aprofundar os seus estudos sobre tabelas, campos e registros, bem como domínio e o que são as chaves primária e estrangeira, assim como o que se trata a normalização.

Ainda verá sobre relacionamentos, modelo entidade-relacionamento, diagrama entidade-relacionamento, condicionalidade, autorrelacionamentos, integridade de banco de dados e regras.

Vamos começar nossa jornada?



Módulo 1

Introdução ao banco de dados

Introdução ao banco de dados

Neste módulo, você estudará os conceitos básicos relacionados ao banco de dados, sua definição, as vantagens e os bancos de dados relacionais. Isso porque os bancos de dados, bem como os sistemas que os administram, são cruciais na atualidade, uma vez que a sociedade se depara quase diariamente em seus afazeres com algum tipo de banco de dados. Tarefas simples, como realizar a reserva de um voo ou de um hotel, acessar o acervo de uma biblioteca on-line e até fazer compras, estão relacionadas a um banco de dados.

Portanto, conhecer as suas principais características, sobretudo, do ponto de vista técnico vai lhe ajudar a compreender o funcionamento de vários sistemas e serviços atuais, como também poderá lhe ajudar durante o desenvolvimento de softwares de diferentes áreas.

Ficou animado(a)? Então vamos lá!

Banco de dados

Para começar, veremos a definição de modelagem de dados. Basicamente, ela é uma metodologia utilizada para determinar as Regras de Negócio (RN) e a arquitetura de um Banco de Dados (BD), ou seja, descreve as estruturas lógicas e físicas do banco.



Entenda o Conceito

Derivada da palavra grega que significa caminho, o termo metodologia é um conjunto de ações e de processos organizados, a fim de se atingir um determinado objetivo.

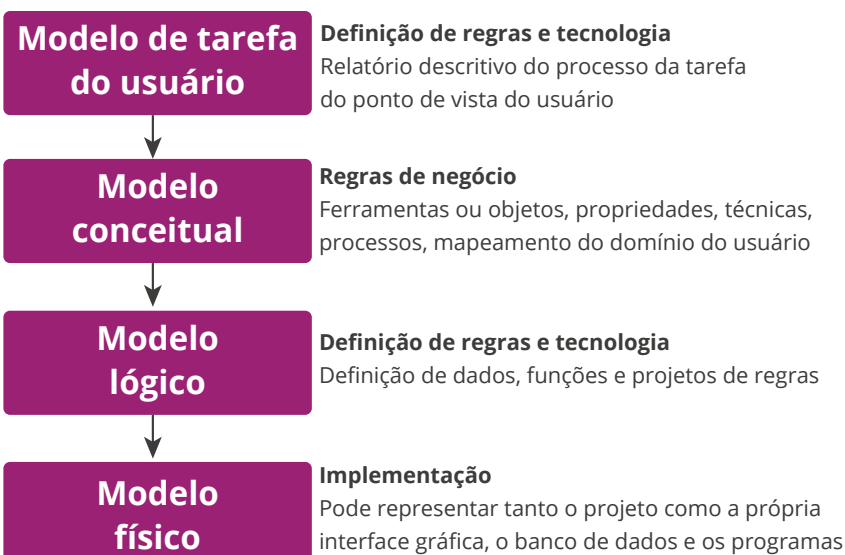
Essa abordagem capta as informações que simbolizam um contexto, estruturando um conjunto denominado **modelo lógico de dados**. Ela provê níveis de abstração de dados que omitem do usuário detalhes sobre a estocagem.

Desse modo, a modelagem de dados pode ser entendida como a representação (abstração) de informações de uma realidade (minimundo), a fim de criar os projetos

conceituais e lógicos de um banco de dados, o qual se divide em: modelo conceitual, modelo lógico e modelo físico.

Observe, abaixo, um exemplo do fluxo de modelagem de dados, considerando esses três modelos.

Do negócio...



... ao sistema

#PraCegoVer: na imagem, temos um esquema de fluxo de modelagem de dados. Do lado esquerdo, encontramos o fluxo desenhado. De cima para baixo, temos o modelo de tarefa do usuário, o modelo conceitual, o modelo lógico e o modelo físico. Do negócio ao sistema. Do lado direito, no modelo de tarefa do usuário, há a definição de regras e tecnologia (relatório descritivo do processo da tarefa do ponto de vista do usuário); no modelo conceitual, há as regras de negócio (ferramentas ou objetos, propriedades, técnicas, processos, mapeamento do domínio do usuário); no modelo lógico, há a definição de regras e tecnologia (definição de dados, funções e projeto de regras); no modelo físico, há a implementação (pode representar tanto o projeto como a própria Interface gráfica, o banco de dados e o programas).

Contudo, atrelada à modelagem de dados, existe o Sistema de Gerenciamento de Banco de Dados (SGBD), ou seja, um sistema que permite fazer alguma alteração nesses dados. Para saber mais sobre isso, convidamos você a ouvir o *podcast* a seguir.



Podcast

Confira o *podcast* sobre SGBD.

Perdeu algum detalhe? Confira o que foi abordado no *podcast*.

Olá! Vamos falar sobre o Sistema de Gerenciamento de Banco de Dados (SGBD)?

Mas antes de chegar nele, vamos lembrar sobre o Banco de Dados (BD), que é conhecido apenas como um recipiente ou repositório para um conjunto de arquivos de dados, como se fosse um armário de arquivo eletrônico. E é aí que o SGBD entra em cena, para organizar e armazenar esses dados sobre um domínio específico. Ou seja, um sistema de gerenciamento de banco de dados representa um sistema computadorizado de manutenção de registros.

Sendo assim, o uso de um SGBD permite executar várias operações com os arquivos, como deletar os que já existem no banco de dados ou criar arquivos novos. Mas também é possível deletar apenas os dados de algum arquivo existente, modificar ou acrescentar dados.

Dentro dessas funções, é possível verificar que a função do SGBD será a de manejar tudo o que ele armazena. Continue conosco para saber sobre mais conceitos da Modelagem de Dados!

Neste tópico, você aprendeu que o conceito básico de modelos de dados, se baseia em uma forma de esquematizar um conjunto de informações abstratas as quais são associadas a elementos da realidade, gerando os projetos conceituais, lógicos e físicos.

Além disso, no *podcast*, você viu sobre o Sistema de Gerenciamento de Banco de Dados (SGBD), cuja função é manejar (deletar, duplicar, modificar etc.) tudo o que está armazenado nele.

É importante que se aprofunde nesse conceito, sobretudo, em relação aos termos minimundo e abstração, os quais o ajudarão no desenvolvimento de softwares.

Minimundo e abstração

Conforme você estudou, o conceito de modelagem de dados é muito importante para os desenvolvedores de programas e, para que isso seja possível, é imprescindível que entenda sobre os termos minimundo e abstração.

Nas etapas iniciais da implementação de um projeto de banco de dados de algum software, são raros os que não precisam de um banco implementado, ocorrendo o levantamento e a análise de requisitos realizados com especialistas do domínio. Para tanto, é feita a coleta de dados por meio de entrevistas, questionários, documentos existentes e validação de procedimentos junto ao cliente (usuário), a fim de se entender e viabilizar a solução sistêmica.

Ou seja, quanto mais detalhes você tiver, maior será o êxito da execução e da conclusão do projeto de desenvolvimento dos programas. Assim, para que isso ocorra, convidamos você a assistir o vídeo abaixo, que traz os conceitos de minimundo e abstração.



Vídeo

Confira o [vídeo](#) sobre minimundo e abstração.

Perdeu algum detalhe? Confira o que foi abordado no vídeo.

Minimundo ou Regras de Negócio (RN) é a criação de um documento com uma descrição textual curta, que acontece a partir de determinadas informações. Nele, é descrito os dados de como as coisas devem funcionar. Ou seja, o minimundo é um primeiro modelo desenvolvido para entender os conceitos da realidade, sem focar no processamento ou na automatização das informações relevantes a essa realidade.

O minimundo, portanto, é um pedaço de realidade do qual o analista levantou certos detalhes, os quais podem ser de interesses para uma área de negócios de determinada organização, por exemplo. Inclusive, ao revelar o interesse em realizar a gestão, seria possível fracionar o minimundo em porções menores, dependendo da sua complexidade, para melhor analisá-lo.

Abstração, por sua vez, refere-se à “visão” que surge mentalmente com base em qualquer realidade. Assim, o ato de abstrair consiste no processo mental utilizado para selecionar características de objetos e situações, sendo que algumas serão excluídas por não apresentarem relevância ao contexto.

Por exemplo, se alguém nos diz para pensar em um cachorro preto da raça labrador, é essa imagem que virá em nossa mente, certo?

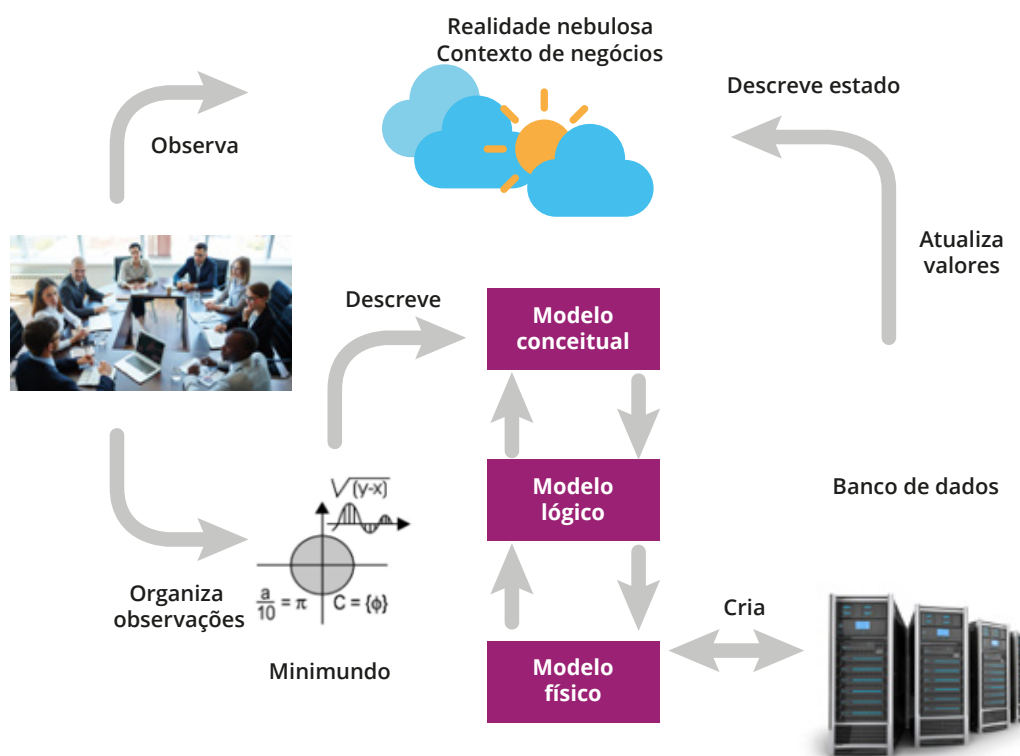
Portanto, a abstração é o modo de retratar os conceitos implícitos nos requisitos (tirados do minimundo) do projeto de banco de dados de um software.

Conclusão: temos o minimundo como uma esquematização da realidade e a abstração como a representação mental dessa realidade.

Continue conosco para saber mais sobre como a determinação de uma realidade é importante na Modelagem de Dados!

Assim, de acordo com o vídeo, o conceito de minimundo, é como uma esquematização, a fim de compreender os conceitos da realidade, principalmente, em relação ao escopo do projeto do software. Já a abstração é uma representação mental da realidade mapeada feita a partir do minimundo.

Em relação a isso, convidamos você a observar com atenção o fluxograma a seguir, em que temos uma aplicação dos dois conceitos apresentados no vídeo.



#PraCegoVer

Na imagem temos um esquema de minimundo e abstração. Da esquerda para a direita, temos que observar a realidade nebulosa e o contexto de negócios, organizando as observações a partir do minimundo, que descreve os modelos conceitual, lógico e físico, em que este último cria um banco de dados. Tal banco atualiza valores e descreve o estado da realidade observada inicialmente.

Note que o fluxograma retrata o que os analistas precisam ter atenção para simbolizar quais as reais necessidades dos dados que os clientes (inseridos em determinado contexto) precisarão, para atingirem os objetivos de seus negócios.

Inclusive, adiante, você poderá se aprofundar sobre o registro das necessidades em relação às informações de alguma realidade.



#PraCegoVer

Na imagem, aparece uma tela com várias linhas de código sobrepostas com cores claras e vibrantes.

Tendo o modelo em mãos, a equipe poderá interagir com os usuários para realizar a homologação de suas metas e seus objetivos, possibilitando detalhar as informações para a base de dados, que possibilitará o desenvolvimento de um sistema de informações (banco de dados), o qual atenda ao que os usuários finais precisam.



#PraCegoVer

Na imagem, há uma programadora sentada em uma mesa de escritório. Em sua frente, existem três monitores de computador, um teclado em que ela digita. Ao lado dela, existem mais dois monitores.

Até aqui, você compreendeu os conceitos de minimundo, uma esquematização de elementos da realidade ligados ao escopo do projeto do software, e de abstração, uma representação mental da realidade mapeada a partir do minimundo. E, com base neles, constatou a relevância de elaborar um modelo de dados, a fim de retratar as necessidades de informações de alguma realidade, ou seja, os objetivos

ligados ao projeto de software a ser feito.

Tendo em vista esses conceitos, é de igual importância que retomemos um conceito abordado anteriormente: o de sistema de gerenciamento de banco de dados. Preparado(a)? Então vamos lá!

Sistema de Gerenciamento de Banco de Dados (SGBD)

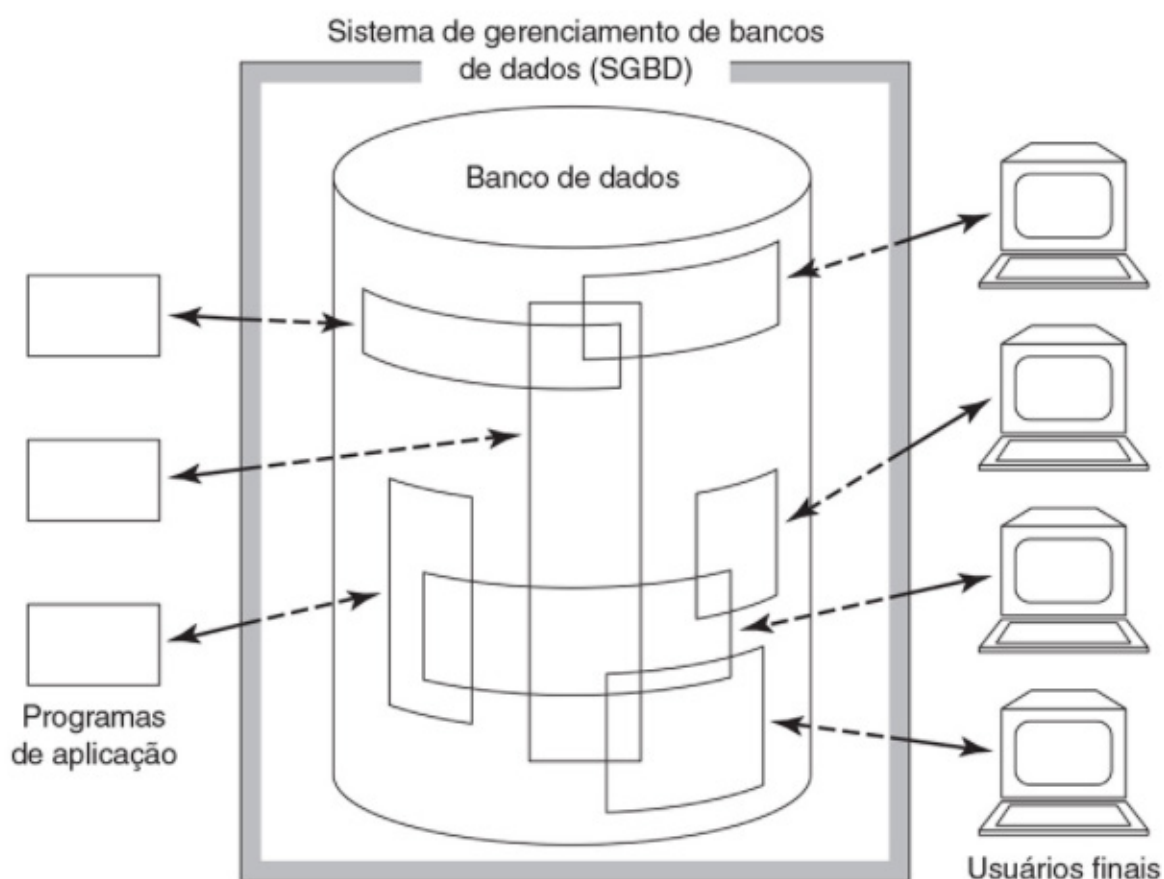
Você se lembra de que não basta existir apenas um banco que armazena os dados os quais serão usados para a construção de qualquer software? Pois bem, também é fundamental que tais dados sejam gerenciados, isto é, atualizados, removidos, adicionados etc.

Assim, para que isso ocorra, é necessário que haja um sistema de gerenciamento de bancos de dados, ou ainda, SGBD. Ele se refere a um sistema informatizado que tem como objetivo o armazenamento de dados, possibilitando que os usuários pesquisem e modifiquem as informações quando forem solicitadas. Essas informações podem ser sobre qualquer coisa indispensável para auxiliar nos processos em geral de uma organização ou para ajudar um indivíduo.

Além disso, no que diz respeito aos dados armazenados, um SGBD traz muitas funcionalidades, como a segurança contra acessos maliciosos ou não permitidos, a proteção contra falhas sistêmicas e de hardware, assim como a distribuição dos dados entre várias aplicações e diversos usuários.

Como exemplos de SGBDs, temos os softwares *OracleDB*, *PostgreSQL*, *MySQL*, *MS SQLServer*, entre outros bastante conhecidos.

Na imagem a seguir, você pode analisar um exemplo de SGBD em que se observa a existência de quatro elementos principais: softwares, dados, usuários e hardware.



#PraCegoVer: na figura, temos um exemplo de sistema de gerenciamento de dados. Trata-se de um cilindro com suas partes, incluindo programas de aplicação e usuários finais.

É essencial que qualquer SGBD tenha, pelo menos, esses quatro elementos. Para conhecer melhor cada um deles, acompanhe a seguir!

Dados

São todas as informações armazenadas em um banco de dados.

Software

Entre os usuários e o banco de dados físico (dados armazenados), há uma camada de software (programas de aplicação), sendo este o SGBD.

Hardware

Refere-se aos volumes de memória secundária (tambores, discos etc.) em que se encontra o banco de dados junto com os dispositivos a ele vinculados, como canais, unidades de controles, entre outros.

Usuários

Existem três tipos principais, estando no topo o programador (usuário que escreve os programas de aplicação que utilizam o banco de dados); seguido do usuário final (tem acesso a um terminal, podendo usar uma linguagem de consulta ou fazer uma chamada para uma aplicação, realizando todas as funções de recuperação, criação, eliminação ou alteração); e, para finalizar, há o usuário que é o administrador do banco de dados, também conhecido por DBA.

Diante disso, você aprendeu mais sobre o conceito de Sistema de Gerenciamento de Banco de Dados (SGBD), o qual garante que seus usuários modifiquem informações, além de proporcionar mais segurança e proteção a tal banco. Como exemplo disso, temos o *OracleDB*, *PostgreSQL* etc.

É importante ter em mente, ainda que você já saiba, o conceito de banco de dados, que é o cerne deste curso. Desse modo, veremos mais detalhadamente sobre ele no próximo tópico!

Banco de dados

Até aqui, é importante que você tenha consolidado o entendimento do termo banco de dados, sobretudo, relacionado à computação. Desta forma, apresentamos abaixo, o conceito básico de banco de dados.



Entenda o Conceito

O banco de dados nada mais é que um repositório para armazenar informações (dados) de qualquer natureza. Ele retrata aspectos do mundo real — ou seja, o conceito visto anteriormente de minimundo — em que qualquer mudança que se faça no minimundo é diretamente replicada no banco de dados.

É habitual citarmos um banco de dados como persistentes, mesmo que não durem por muito tempo. Isto é, quando se menciona que o banco é persistente, intuitivamente se sugere que as informações contidas nele se distinguem quanto a tipos de dados temporários, ou seja, qualquer dado que contenha uma natureza momentânea (breve). Temos como exemplo os dados de saída, entrada dos blocos de controle de software, filas de trabalho, resultados intermediários e instruções SQL.



Atenção

Os dados persistem quando são aceitos pelo SGBD para a inserção no banco. Assim, somente será possível removê-los do banco, posteriormente, por meio de alguma solicitação explícita ao SGBD.

Com base nesse conceito de persistência, é possível prover uma definição precisa sobre o banco de dados, que seria um conjunto de dados persistentes, utilizados pelos sistemas de determinada organização. Esta, inclusive, geralmente necessita armazenar vários dados insistentes quanto à sua operação, como aqueles sobre produtos, contas, planejamento, alunos, pacientes, funcionários etc.

Os bancos de dados podem ser **categorizados** de várias maneiras, sendo que uma das principais delas aponta para a sua arquitetura, a qual pode ser centralizada, descentralizada, distribuída e replicada. Veja na sequência e descubra sobre cada uma delas!

Centralizada	Arquitetura em um servidor em que as aplicações ou os clientes podem acessar o banco de dados. Logo, a responsabilidade pela capacidade de armazenamento e resposta será assumida. Para tal modelo, é necessário alto poder de processamento do servidor e um ótimo desempenho do SGBD.
Descentralizada	Há mais de um servidor para o banco de dados, possibilitando que eles sejam descentralizados. Tem-se, então, garantia de autonomia local e auxílio na comunicação, que, por exemplo, pode ser melhor distribuída entre as aplicações que utilizam tabelas específicas.
Distribuída	Arquitetura em que os dados estão compartilhados em muitos computadores ou servidores, com atualizações ou sincronismo para se certificar da integridade desses dados nos locais em que se encontram.

Replicada

Arquitetura em que o banco de dados é copiado para muitos computadores ou servidores, como em uma metodologia de espelhamento. Todos os bancos são idênticos e, conforme há alterações no primeiro banco, os demais são alterados em cascata. Aliás, esse tipo de arquitetura garante a segurança das informações. No caso de haver falhas em algum *host*, outro poderá assumir seu lugar, visto que os dados são idênticos.

Viu como é simples o conceito de banco de dados, sobretudo, no contexto da computação? Assim, foi possível entender que esse conceito está ligado a um repositório que armazena informações (breves, ou não) sobre a realidade, possuindo uma arquitetura própria que categoriza os dados. Inclusive, você entendeu que essa arquitetura pode ser centralizada, descentralizada, distribuída ou replicada.

A seguir, daremos passos rumo às etapas de construção dos modelos de dados, sendo que a primeira etapa que veremos é o modelo conceitual. Acompanhe o conteúdo!

Modelo conceitual

O modelo conceitual utiliza a análise de requisitos por base, sendo um grupo de hipóteses que leva em conta o mundo real, indicando as regras de negócio do sistema. Desse modo, consiste no detalhamento de alto nível, voltado a entender e descrever todos os detalhes de uma organização. Esse detalhamento das informações de negócio, por sua vez, é guardado no banco de dados.

Com isso, convidamos você a assistir o vídeo a seguir para conhecer mais detalhadamente sobre o modelo conceitual!



Vídeo

Confira o [vídeo](#) sobre o modelo conceitual.

Perdeu algum detalhe? Confira o que foi abordado no vídeo.

Olá! Aqui falaremos com mais detalhes sobre o modelo conceitual na Modelagem de Dados.

Vamos começar pelo seu objetivo, que é a implementação de um sistema com informações coerentes referentes aos elementos que o compõem, como seus relacionamentos, suas propriedades e seus objetos. De modo geral, tais elementos precisam estar bem estruturados para que o usuário entenda o seu propósito.

É importante frisar que esse modelo não foca em questões como a abordagem do banco de dados que será escolhida, nem em como será acessado ou quais estruturas físicas serão desenvolvidas pelo SGBD.

A representação gráfica do modelo retrata determinada realidade em um contexto e suas estruturas de dados, observe.



Com a inexistência desse modelo, seria difícil ter um bom entendimento das regras de negócio, o que acarretaria a possibilidade de criação de sistemas que não atendam à necessidade do cliente ou a finalidade para a qual foi criada.

Apartir desse estudo você pôde observar a primeira etapa de construção do modelo conceitual. No próximo tópico, continue seus estudos da modelagem de dados.

Aqui, você compreendeu sobre a primeira etapa da construção de um modelo de banco de dados, o modelo conceitual, cuja função é implementar um sistema que possua informações precisas e alinhadas ao projeto de desenvolvimento de um programa específico, atendendo, assim, à finalidade do cliente.

Sabendo disso, vamos para a segunda etapa da modelagem de dados. Preparado(a)?

Modelo lógico

A etapa de criar um modelo lógico somente inicia quando o modelo conceitual estiver concluído. Desse modo, apresenta questões referentes às possibilidades de abordagens tecnológicas do SGBD, para criar a lógica dos relacionamentos que há no modelo conceitual.

Assista a mais um vídeo para entender esse processo e as características do modelo lógico. Acompanhe!



Vídeo

Confira o [vídeo](#) sobre modelo lógico.

Perdeu algum detalhe? Confira o que foi abordado no vídeo.

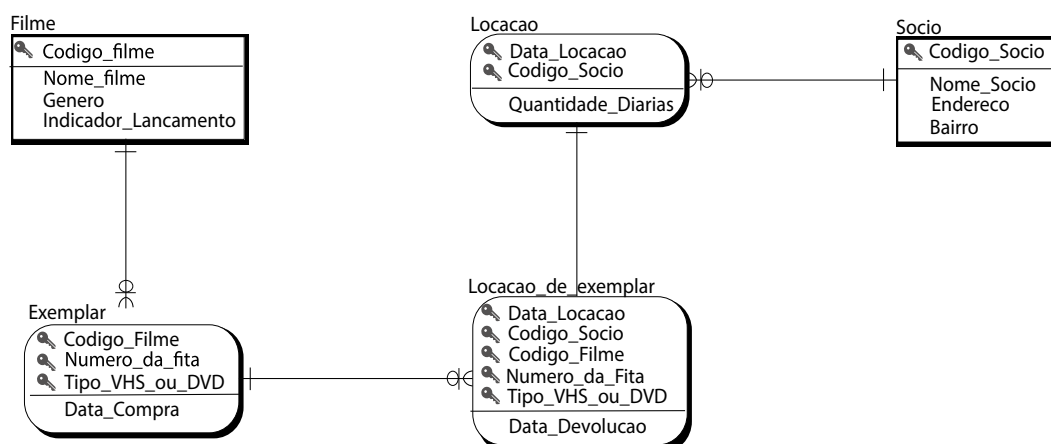
Olá! Vamos falar sobre modelo lógico?

Ele detalha a forma lógica das estruturas, como relações, entidades, tipo de dados, hierarquia, entre outras particularidades. Estes elementos existem no banco de dados conforme a possibilidade de sua abordagem, sem levar em conta os detalhes de determinado SGBD.

A representação gráfica do modelo lógico é uma espécie de esquema lógico em relação à uma abordagem, observe. O modelo descreve quais tabelas, campos e chaves primárias e estrangeiras o banco terá.

Vamos a um exemplo!

Aqui temos que “Filme”, “Locacao”, “Exemplar”, “Socio” e “Locacao_de_exemplar” serão os nomes das tabelas, enquanto os dados contidos (“Nome_Filme” e demais) serão as variáveis (campos) que armazenarão essas informações.



Como nesta parte, onde o “Codigo_Filme” da tabela “Filme” será a chave primária, mas, nas demais tabelas, esse mesmo dado será uma chave estrangeira para referenciar a tabela a qual o filme está vinculado — por isso o ícone de chave.

Em resumo, é como se fosse uma foto da estrutura que a tabela terá dentro do banco de dados.

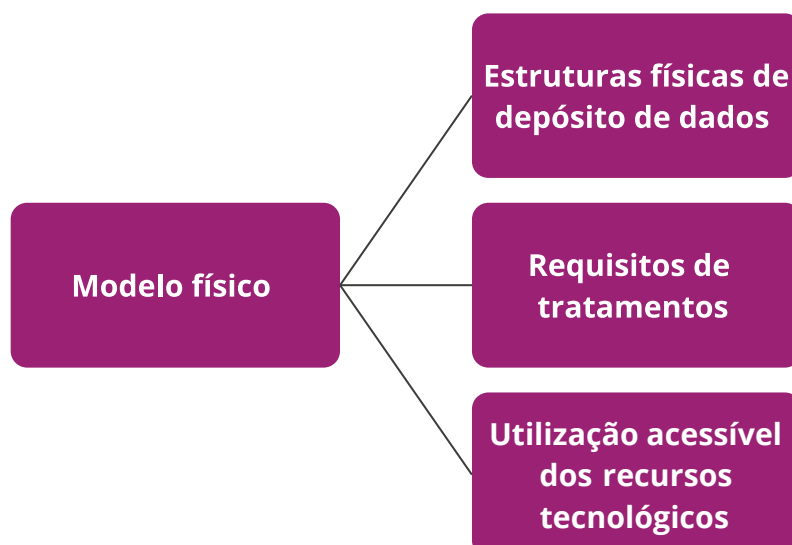
No vídeo você compreendeu a estrutura dos elementos do modelo lógico (tipos de dados, hierarquia, quais são as classes etc.) por meio de um esquema gráfico de chaves que são articuladas entre si.

Todavia, ainda existe a última etapa, que é o modelo físico, o qual você acompanhará a seguir.

Modelo físico

Após você compreender os modelos conceitual e lógico, chegou a hora do modelo físico, a última etapa de construção do banco de dados.

O modelo físico só será criado quando tivermos o modelo lógico, com o detalhamento das estruturas físicas de depósito dos dados, como tipos e tamanhos dos campos. Geralmente, esse modelo é projetado seguindo os requisitos de tratamentos, considerando a utilização acessível em relação aos meios de recursos tecnológicos disponíveis para o projeto.



#PraCegoVer: na imagem temos um esquema do modelo físico, que é construído somente depois do modelo lógico, com estruturas físicas de depósito de dados, projetado conforme os requisitos de tratamentos e com utilização acessível quanto aos recursos tecnológicos disponíveis.

Nesse momento, é necessário ter atenção na hora de criar as estruturas de dados físicas no SGBD, pois terão a responsabilidade de armazenar as informações do banco e auxiliar na escolha do SGBD apropriado ao sistema. Dessa forma, incluindo as características ou opções que provêm um melhor relacionamento, de acordo com os recursos e modelos disponíveis.

O modelo físico consiste na fase final, ou seja, aquela que antecede a criação de banco de dados, considerado como um aperfeiçoamento do modelo lógico.



Saiba mais

Convidamos você a assistir ao vídeo [Modelagem de Dados - Modelo Conceitual, Lógico e Físico](#), que traz informações complementares aos seus conhecimentos! Vale a pena conferir!

Assim, você entendeu mais acerca da última etapa da modelagem de dados: o modelo físico. Ele representa as estruturas físicas de depósito dos dados, evidenciando os diferentes tipos e tamanhos dos campos, sem desconsiderar o escopo do projeto e a tecnologia disponível.

No próximo tópico, você entenderá mais acerca dos benefícios da utilização desses bancos, principalmente, quando estiver atuando na construção de softwares. Confira!

Vantagens de utilizar um banco de dados

Após aprender de modo detalhado as três etapas da modelagem de dados (modelo conceitual, lógico e físico), neste momento, iremos esmiuçar as vantagens de um sistema de banco de dados, aspecto muito relevante para a sua futura atuação profissional.

Quando comparado aos métodos tradicionais que utilizam o papel para registrar e manter dados, o sistema de bancos de dados produzido no computador apresenta uma série de benefícios, os quais você pode conferir adiante.

Velocidade

O sistema obtém e atualiza informações em geral de forma mais rápida que um ser humano, por exemplo.

Densidade

Não é necessária a existência de arquivos de papel, que provavelmente ocupariam espaço.

Trabalho menos monótono

A maior parte do tédio para realizar a manutenção manual em arquivos é abolida, sendo que as tarefas sistêmicas são, geralmente, realizadas com maior qualidade pelas máquinas.

Melhora de relacionamentos e produtividade

Os setores se comunicam melhor, dispondo de expectativas, metas e objetivos claros em relação à gestão de dados.

Atualidade

Dados precisos e atualizados estão à disposição para consulta.

Proteção

Os dados ficam protegidos contra danos não intencionais e acessos não autorizados.

Redução de riscos na operação

Uma vez que haverá transferência nas operações, ocorre a redução de risco, visto que as áreas se mantêm alinhadas, realizando entregas assertivas e melhorando a performance.

Mais precisão nas decisões

As organizações que recorrem ao banco de dados apresentam transparência e confiança para decidirem questões importantes, podendo utilizar os dados para melhor avaliarem o cenário.

As vantagens apresentadas são de aplicabilidade em escala maior nos ambientes de sistemas multiusuário, ou seja, em que o sistema de banco de dados será acessado por vários usuários simultaneamente. Nesse caso, a base de banco de dados provavelmente será mais complexa quando comparada a de um ambiente monousuário, cujo acesso é por um usuário e de uma base mais simples.

E por falar em vantagens, você conhece aquelas envolvidas na **abordagem de banco de dados** relacionadas às noções de controle centralizado? Caso ainda não conheça, detalharemos mais sobre isso ao longo desse tópico, sendo que o primeiro passo é ouvir o *podcast* a seguir.



Podcast

Confira o *podcast* as vantagens de Banco de Dados.

Perdeu algum detalhe? Confira o que foi abordado no *podcast*.

Olá! Vamos conferir algumas vantagens de um banco de dados?

A primeira delas é o **compartilhamento de dados**. Esta partilha não é apenas em relação aos sistemas existentes, mas sim para que possam compartilhar informações de banco de dados.

Ela se refere, também, aos sistemas que poderão ser desenvolvidos para operarem com a mesma base de dados. Assim, é possível atender aos requisitos de dados dos novos sistemas, sem a necessidade de haver novos dados adicionados ao banco.

A segunda vantagem é a **redução da redundância quando necessário**. Em SGBD, cada aplicação tem seus arquivos privados, podendo, com isso, haver um número considerável de informações redundantes (repetidas), levando a um desperdício no espaço de armazenamento. Mas, por meio da integração de arquivos, pode-se reduzir essas redundâncias.

Há casos específicos, de acordo com o negócio, que necessitam de cópias dos mesmos dados, mas, em geral, as repetições deverão ser controladas com atenção. Em suma, o SGBD deverá ter conhecimento e responsabilidade em disseminar as atualizações.

Nesse caso de redundâncias que não são removidas, mas controladas, temos a terceira vantagem, a qual o SGBD garantirá que um banco de dados dificilmente se torne inconsistente para o usuário, certificando-se de que as mudanças em uma de suas entradas sejam replicadas automaticamente para a outra entrada, realizando a propagação das atualizações.

A quarta vantagem é que poderá ser fornecido um suporte às transações. Uma transação se refere a uma unidade lógica de trabalho de banco de dados que, normalmente, contém várias operações desses bancos, principalmente as relacionadas às alterações. Por exemplo, uma das possibilidades de suporte em transações destaca a transferência bancária de um valor localizado na conta Y para a conta X. Tal situação exige, pelo menos, duas operações:

- uma para retirar da conta Y;
- outra para depositar na conta X.

Entretanto, caso o usuário declare que as duas alterações pertencem à mesma transação, o sistema garantirá que ambas sejam executadas ou nenhuma delas, no caso de o sistema falhar devido a uma queda de energia, por exemplo.

Por fim, temos a última e quinta vantagem, que é a preservação da integridade. Isso se deve, porque, observando o controle centralizado aplicado ao banco de dados, ainda que não existam inconsistências

entre as duas entradas que deveriam conter o mesmo valor, o banco poderá ter informações incorretas, como um produto com nome que não existe ou um funcionário com 800 horas de trabalho em um dia em vez de oito horas.

Viu só, como o controle centralizado permite evitar problemas com informações equivocadas, sendo estas cinco vantagens importantíssimas para o contexto do banco de dados?

O controle centralizado evita problemas de informações equivocadas, uma vez que autoriza a inserção de restrições de integridade determinadas pelo administrador de dados e desenvolvidas pelo Analista de Banco de Dados (DBA).

Por conta disso, além de detalharmos os benefícios, também é importante abrir espaço para uma vertente ligada às vantagens do controle centralizado aplicado ao banco de dados. Sobre isso, podemos citar três, observe quais são elas.



#PraCegoVer

Na imagem temos uma fotografia editada digitalmente. Uma pessoa está com as mãos estendidas para frente. No centro, há um holograma de cadeado com linhas e conexões ao redor, retratando a segurança em softwares e programações.

A primeira é **reforçar a segurança**.

Sabendo que o DBA terá acesso completo ao banco, com o auxílio da administração do banco de dados, pode-se verificar esse acesso e se ele ocorrerá por canais apropriados, como o formulário de uma requisição no sistema de solicitações. Dessa forma, é possível determinar as restrições de segurança para cada tipo de acesso, por exemplo de exclusão, busca, alteração etc., as quais serão acionadas quando existir a tentativa de acessos não autorizados de certos perfis.

A segunda são os **requisitos contraditórios que poderão ser equilibrados**. Uma vez que o DBA conhece os requisitos globais da organização, em oposição aos requisitos dos usuários individualmente, poderá arquitetar o sistema de maneira a prover um serviço global, que seja o ideal para a organização.



#PraCegoVer

Uma pessoa digitando em um notebook, ao fundo existe uma outra pessoa em outro computador realizando alguma atividade laboral.



#PraCegoVer

Uma pessoa segurando um tablet. Perto de uma das extremidades ela está pressionando a superfície com o dedo indicador, Enquanto que a outra mão segura o tablet. Acima dele, há uma imagem projetada de um ícone de engrenagem com um símbolo de correto dentro.

Por fim, em relação à última, temos que a abordagem de banco de dados possibilita a **padronização**. Assim que o DBA dispõe de um controle central do banco de dados, garantirá que os padrões aplicáveis sejam atendidos na representação dos dados, como nos padrões relacionados a departamentos, instalação, organização etc.

Porém, após conhecer as vantagens associadas ao uso de um banco de dados feito no computador, bem como o compartilhamento de dados e a diminuição da redundância quando necessário, é imprescindível que você entenda mais um

conceito muito relevante: **o banco de dados relacional**. Desse modo, para que isso seja feito, apresentamos uma breve história desse conceito, a qual se inicia com a criação da Teoria Relacional.

Ela foi criada pelo matemático Edgard Frank Codd, em 1970. Nove anos mais tarde, em parceria com o pesquisador e consultor Christopher Date, o projeto foi aperfeiçoado, transformando-se em um modelo relacional estendido em 1979. O modelo relacional, então, constitui um modo de detalhar o banco de dados a partir de conceitos matemáticos, como a Teoria dos Conjuntos.

Com o foco no aperfeiçoamento da visão dos dados pelos usuários, a **Teoria Relacional** permite que usuários enxerguem o banco de dados como se fosse um grupo de tabelas bidimensionais, as quais contêm linhas e colunas.

Todavia, é preciso se debruçar ainda mais em relação aos conceitos correlatos a Teoria Relacional, sobretudo, o banco relacional. Com base nisso, nos próximos tópicos, apresentamos os principais conceitos de um banco de dados relacional. Confira o conteúdo!

Tabelas, campos e registros

Vamos agora detalhar os conceitos correlatos, principalmente, associados ao banco de dados. Tais conceitos são o de tabela, colunas e linhas.

Nos conceitos relacionais, a tabela (**relação**) diz respeito a um grupo de dados organizados em colunas (**atributos**) e linhas (**tuplas**).

As colunas simbolizam um atributo de dados (campos de uma tabela), enquanto as linhas simbolizam o valor desse atributo (valor que o campo terá), sendo que, juntos, geram o registro de uma tabela. Logo, cada linha de uma tabela consiste no registro composto por um grupo de campos com informações.

De acordo com a Teoria Relacional, há algumas regras que determinam uma tabela de dados. Confira a seguir!

Relação

É como cada tabela é referenciada.

Tupla

Refere-se ao grupo de linhas que a compõem a tabela.

Colunas

Serão nomeadas e retratarão um domínio da tabela.

Outros pontos, também, merecem atenção, por exemplo:

Disposição

Não importa a disposição (organização) das linhas, pois não existirão linhas idênticas.

Nomeação

É necessário utilizar nomes para referenciar as colunas, não importando sua disposição (ordem).

Nomes diferentes

Não haverá tabelas com nomes iguais no mesmo banco de dados, sendo que cada uma terá um nome diferente.

Entidade

Outro conceito é a entidade, que se refere a qualquer objeto, coisa ou pessoa abstraído de alguma realidade para pertencer a uma tabela e guardar seus dados.

Componentes e valores

As colunas de uma tabela se referem aos seus componentes e valores, os quais devem ser indivisíveis. Isso significa que não há colunas da espécie subgrupo, pois são elementos importantes que não autorizam a ocorrência múltipla de valores.

Agora que você já compreendeu os conceitos de tabela (relação), colunas (atributos) e linhas (tuplas), está na hora de entender sobre o domínio. Para isso, acompanhe o próximo tópico.

Domínio

O conceito domínio, consiste em ser um grupo de valores, geralmente infinitos, que são determinados e nomeados, cujos atributos assumem os valores contidos. É possível defini-lo separadamente das entidades, as quais permitem sua reutilização e padrão.

Em outras palavras, o domínio é um grupo com valores atômicos referentes a uma coluna e sua tabela (relação). Veja alguns exemplos:

Aniversário

Grupo de oito dígitos.

Celular

Grupo de 11 dígitos.

RG

Grupo de 10 dígitos.

Setor

Grupo de setores de uma organização.



#PraCegoVer: na imagem temos uma fotografia editada digitalmente. Uma pessoa está sentada diante de uma mesa e escreve em uma agenda com a mão esquerda. Sua mão direita segura um celular. Dele, surgem valores distintos de dados. Em cima da mesa, há parte de um notebook, uma caneta e uma xícara.

Existe, ainda, uma associação para cada domínio, a qual é referente ao tipo de informação ou formato, veja os exemplos abaixo:

Idade

Número inteiro entre 18 e 75.

celular

(XXX) XXXXX-XXXX, em que $X = \{0,1,2,3...9\}$.



Na prática

As restrições de domínio consideram por base a definição dos domínios de valores para todas as colunas de uma tabela. Geralmente, estabelecem-se e determinam os valores que uma tabela terá, resultando no domínio da coluna.

É comum o domínio ser definido por tipos de dados padrões, por exemplo: *integer* (número inteiro), *float*, *date*, *char*, *time*, entre outros.

Dessa maneira, podemos dizer que o conceito de domínio no contexto de banco de dados, é a representação de valores ligados a uma coluna de uma tabela, tal qual um agrupamento deles. A partir disso, esses valores são definidos e nomeados, de acordo com os atributos do banco de dados.

Só que existe uma possibilidade, dentro do banco de dados, de que as linhas que formam a tabela possam se repetir, ou até mesmo não serem reconhecidas durante o seu uso. Porém, a fim de que isso não ocorra, entra em cena a chave primária, a qual veremos em mais detalhes a seguir.

Chave primária

A organização do banco de dados é muito importante, ainda mais diante de situações específicas, a exemplo da repetição de informações. Assim, para garantir que isso não ocorra, utiliza-se a propriedade de chave primária ou *Primary Key* (PK).

Normalmente haverá uma coluna ou um grupo de colunas em que jamais se repetirá determinado valor em outra linha. Assim, ao escolher um campo para ser a chave primária, será passada a informação ao banco de dados de que não deverá haver dois registros com valores iguais. Assim como no exemplo a seguir.



Na prática

Se definimos o campo de matrícula na tabela de aluno como chave primária, será passada a informação ao banco de dados de que não deve haver alunos com a matrícula igual, pois será uma informação única.

Logo, uma tabela pode ter apenas uma chave primária, que costuma ser referenciada como COD, ID ou algo do gênero. De modo geral, ela será criada quando uma tabela possuir os seguintes propósitos:

Unidade

Que um registro seja único.

Valor

Que não seja nulo o valor do registro.

Identificação

Que o registro possibilite a identificação da tabela.

Além de seguir os propósitos mencionados, a chave primária pode ser categorizada em simples ou composta. Você sabe a diferença entre essas categorias? Acompanhe!

Chave simples

É constituída de um único campo da tabela em que não terá dois valores ou o mesmo valor para mais de um registro. Este não poderá ser nulo.

Chave composta

É constituída por mais de um campo da tabela em que os valores poderão se repetir, mas não poderá haver combinações desses valores.

Assim, uma chave primária reconhece uma tabela a partir da singularidade de um registro e imposição de não ser valor nulo. Portanto, ela tem por objetivo a identificação de um registro.

A chave primária é um recurso muito importante para a organização do banco de dados, sobretudo, em relação a evitar repetições e outros entraves no momento de se manejar o repositório de informações. Além disso, para que ela seja criada, deve-se atender aos três propósitos, bem como que a chave primária pode ser do

tipo simples, ou composta.

Porém, no contexto de banco de dados, existem situações que vão além de uma tabela apenas, envolvendo duas ou mais, fazendo com que a chave primária seja ineficaz. Assim, é o momento em que a chave estrangeira atua, inclusive, será tema do próximo tópico. Acompanhe com atenção!

Chave estrangeira

A chave estrangeira faz referência à ligação entre diferentes tabelas, isto é, quando há uma chave primária de uma tabela que pertence, igualmente, aos campos de outra tabela, ocorre a referenciação de um componente em relação a outro de uma segunda tabela, definindo um relacionamento lógico.

Nesse sentido, é válido ressaltar que a ligação entre tabelas por meio da chave estrangeira tem dois objetivos, os quais você pode ver a seguir.

Valor

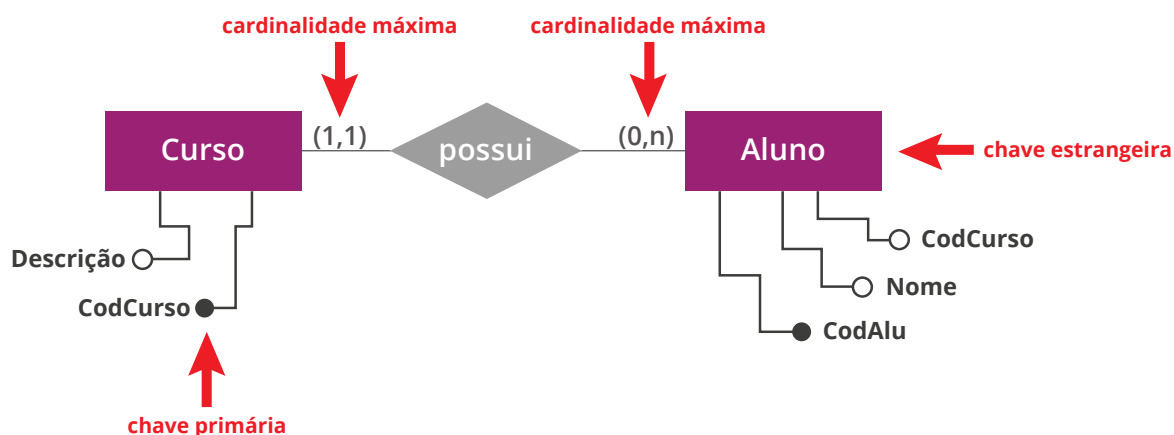
Impossibilitar que seja adicionado no ID de uma tabela um valor não válido.

Registro

Impossibilitar que seja deletado um registro referenciado em outra tabela.

Logo, a chave estrangeira, também conhecida por *Foreign Key* (FK), tem por objetivo principal identificar o relacionamento entre as tabelas.

Na imagem a seguir, observe um exemplo de chave primária e estrangeira.



#PraCegoVer: na imagem, temos exemplos de chaves primária e estrangeira. A chave primária está retratada à esquerda, em que há um retângulo com o texto "Curso". Dele, saem duas linhas, uma de descrição e outra de "CodCurso", que é a chave primária. A cardinalidade é máxima (1, 1). O "Curso" possui "Aluno". Este traz cardinalidade máxima também (0, n). Do lado direito, temos um retângulo com o texto "Aluno". Dele, saem três linhas, uma de "CodAlu", uma de nome e outra da "CodCurso", que seria a chave estrangeira.

Neste ponto, é importante destacar que, ao contrário da chave primária, a chave estrangeira possui quantas chaves forem necessárias para poder referenciar as ligações em outras tabelas.

Saiba mais



Em seu futuro profissional, quando você receber uma demanda voltada à construção de um banco de dados, é importante que seja realizado um projeto para esse fim, sendo que o uso da abordagem de uma metodologia adequada auxiliará no sucesso dele.

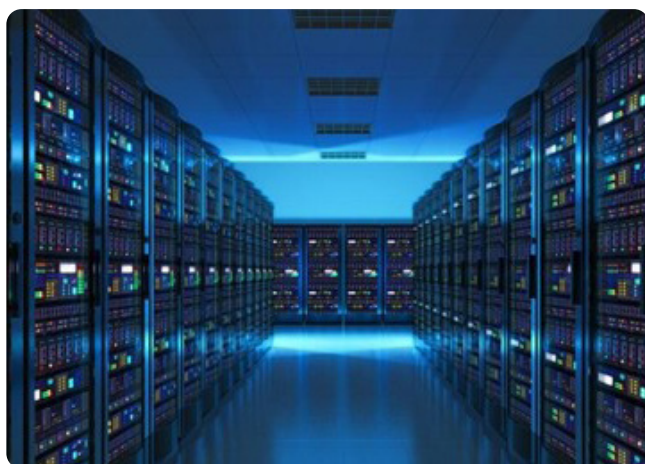
Para complementar seus conhecimentos sobre como elaborar o projeto tendo em vista esses princípios, convidamos você a assistir ao vídeo Bancos de Dados - Projeto de [Banco de Dados](#) e ler o texto sobre [Projeto de Banco de Dados](#). Vale a pena conferir!

Neste tópico, você aprendeu sobre o conceito de chave estrangeira e os dois objetivos que ela atende. Contudo, para que não só a chave estrangeira, mas todos os outros elementos que compõem o projeto tenham êxito e funcionem

ordenadamente, é relevante entender mais um conceito, o de normalização, o qual veremos a seguir!

Normalização

A normalização consiste no conjunto de regras que buscam a organização de um projeto de banco de dados, a fim de reduzir a quantidade de informações, assim como ampliar a integridade e o desempenho. É, portanto, uma metodologia que apresenta os passos para revisar os atributos de alguma entidade, com a finalidade de impedir anomalias encontradas em rotinas de exclusão, alteração ou inclusão de registros no banco.



#PraCegoVer

Na imagem, existe um corredor e que vai em direção a uma parede que contém servidores e que se bifurca em um caminho que vai para direita e outro para a esquerda. Nesse corredor, nas duas paredes, também existem servidores instalados verticalmente.

Podemos entender que a normalização impede que possíveis falhas no projeto de banco surjam, auxiliando na eliminação de mescla de assuntos e redundâncias desnecessárias referentes à essa mistura.

Com isso, várias regras são aplicadas nas tabelas para validar se elas estão estruturadas de acordo com o projeto de banco. Ao todo, são cinco regras de normalização, mas, na prática, apenas três são utilizadas: 1FN, 2FN e 3FN. Vamos ver mais detalhadamente cada uma delas. Confira!

A **primeira forma normal ou 1FN** é uma tabela com todas as colunas compostas por um único valor, não contendo algum grupo de colunas repetidas com atributos compostos ou em uma linha. Assim, utilizar a 1FN consiste em remover da estrutura todos os itens repetidos. Veja um exemplo:

Exemplo de 1FN: Clientes

Tabela de Clientes: Estrutura original

Código	Nome	Telefone	Tipo_tel	Rua	No	Cidade
00001	Maria	3441 8566	Residencial	Contorno	2316	Belo Horizonte
00001	Maria	3215 8751	Serviço	Contorno	2316	Belo Horizonte
00001	Maria	9158 3239	Celular	Contorno	2316	Belo Horizonte
00002	Antônio	8874 5698	Celular	Afonso Pena	5693	Belo Horizonte

Estrutura Normalizada na 1FN:

Tabela Clientes

Código	Nome	Rua	No	Cidade
00001	Maria	Contorno	2316	Belo Horizonte
00002	Antônio	Afonso Pena	5693	Belo Horizonte

Tabela: Telefones_Clientes

Código	Telefone	Tipo_tel
00001	3441 8566	Residencial
00001	3215 8751	Serviço
00001	9158 3239	Celular
00002	8874 5698	Celular

#PraCegoVer: na imagem temos três tabelas, estando uma acima e duas abaixo, uma do lado da outra. A primeira tabela (1FN), na parte superior, é de clientes, com a estrutura original, incluindo dados como código, nome, telefone, tipo de telefone, rua, número e cidade. A segunda tabela, abaixo, à esquerda, é a estrutura normalizada da 1FN de cliente, com código, nome, rua, número e cidade. Já a terceira tabela, à direita, é a estrutura normalizada da 1FN de telefone dos clientes, incluindo código, telefone e tipo de telefone.

A **segunda forma normal ou 2FN** é uma tabela classificada como 1FN e que contém atributos que não participam da chave primária, mas dependem dela. Dessa forma, aplica-se a 2FN, que seria a remoção dos atributos. Com uma chave composta, ou seja, uma chave primária constituída por mais de um campo, todos os dados dependem apenas de alguma de sua parte.

Para validar os atributos dependentes da chave primária e remover da relação os atributos de um grupo independente, será criada uma relação que terá o atributo, não sendo uma chave. Observe o exemplo:

Exemplo de 2FN: Empregados trabalhando em projetos

Tabela de Empregado_Projeto: Estrutura original

Num_emp	Num_proj	Horas	Nome_emp	Nome_proj	Local_proj
00001	001	8	Maria	Versão Evolutiva 3.22	João Monlevade
00002	001	18	José	Versão Evolutiva 3.22	João Monlevade
00003	002	12	Samara	Versão Evolutiva 3.21	Belo Horizonte

Estrutura Normalizada - 2FN

Tabela: projetos

Num_proj	Nome_proj	Local_proj
001	Versão Evolutiva 3.22	João Monlevade
002	Versão Evolutiva 3.21	Belo Horizonte

Tabela: Empregado_projeto

Num_emp	Num_proj	Horas
00001	001	8
00002	001	18
00003	002	12

Tabela: Empregado

Num_emp	Nome_emp
00001	Maria
00002	José
00003	Samara

#PraCegoVer: na imagem, temos quatro tabelas, uma embaixo da outra, com as duas últimas uma ao lado da outra. A primeira tabela (2FN) é de empregado em projeto, em sua estrutura original, incluindo número do empregado, número do projeto, horas, nome do empregado, nome do projeto e local do projeto. A segunda tabela é a estrutura normalizada de projetos, com número do projeto, nome do projeto e local do projeto. A terceira tabela, à esquerda, indica os empregados em projeto com número do empregado, número do projeto e horas. Já a quarta tabela, à direita, é de empregados, com número e nome.

Já a **terceira forma normal ou 3FN** é uma tabela classificada com 2FN e sem campos que dependem de outros tidos como não chaves. Em outras palavras, não pode haver atributos que não são chaves ligados a outros atributos não chaves, pois pode ocasionar anomalias de exclusão, inclusão ou alteração.

Sendo assim, aplica-se a 3FN, que consiste em remover da estrutura esses campos que dependem de outros campos que não são chaves.

A remoção dos atributos independentes de chave primária e criação de uma nova ligação, terá o grupo de atributos e determinará com a chave primária quais serão diretamente dependentes. Veja na prática:

Exemplo de 3FN: Empregados trabalhando em departamentos

Tabela de Empregado_Depto: Estrutura original

Num_emp	Nome	Data_nasc	Num_Depto	Nome_Depto	Emp_Ger_Depto
00001	Maria	06/03/1977	001	Homologação	018
00002	José	27/05/1973	002	Homologação	018
00003	Samara	24/08/1984	003	Desenvolvimento	005

Tabela: Empregado_Depto

Num_emp	Nome	Data_nasc	Num_Depto
00001	Maria	06/03/1977	001
00002	José	27/05/1973	002
00003	Samara	24/08/1984	003

Tabela: Departamento

Num_Depto	Nome_Depto	Emp_Ger_Depto
001	Homologação	018
002	Homologação	018
003	Desenvolvimento	005

#PraCegoVer: na imagem, temos três tabelas, uma embaixo da outra. A primeira tabela (3FN) é de empregados por departamento, em sua estrutura original, incluindo número do empregado, nome, data de nascimento, número do departamento, nome do departamento e empregados gerais por departamento. A segunda tabela indica a estrutura de empregados por departamento resumida, com número do empregado, nome, data de nascimento e número do departamento. Já a terceira tabela indica os dados do departamento, com número, nome e empregados gerais.

É importante frisar que a normalização, nesse caso de 3FN, deve ser aplicada em um relacionamento por vez, visto que, no decorrer do processo, serão obtidas quebras e, conseqüentemente, novos relacionamentos.

Parabéns! Você finalizou o módulo 1.

Nele, você estudou os conceitos fundamentais sobre a modelagem de banco de dados, principalmente, a definição de banco de dados, bem como os conceitos de minimundo, de abstração, os modelos conceitual, lógico e físico, as vantagens de utilização do banco de dados, os conceitos de tabelas, campos, registros, domínio, chave primária, chave estrangeira e normalização.

Vale ressaltar que dominar esses conhecimentos é essencial para a prática de modelagem de banco de dados.

No próximo módulo, a parte prática desses conceitos será mais explorada, sobretudo, no que diz respeito ao relacionamento de um banco de dados.

Pronto para iniciar? Então vamos lá!



Módulo 2

Relacionamentos

Relacionamentos

No segundo módulo, apresentaremos os principais conceitos sobre os bancos de dados relacionais, que tem por finalidade armazenar e prover acessos de dados relacionados. O modelo relacional foi criado para sanar dificuldades com dados desnecessários, provendo um padrão mundialmente seguido para buscar dados que poderiam ser utilizados por qualquer sistema, de modo eficiente, intuitivo e flexível, a fim de guardar e acessar os dados estruturados.

Diante disso, iniciaremos nossos estudos conhecendo o primeiro modelo relacional que se tem notícia, o Modelo Entidade Relacionamento, ou também conhecido pela sigla MER.

Modelo Entidade Relacionamento (MER)

O Modelo Entidade Relacionamento (MER) foi criado em 1976 como uma maneira de unificação das visões de um banco de dados relacional, tendo por base a Teoria Relacional, desenvolvida em 1970.

Reconhecido como MER, o modelo recebeu vários aperfeiçoamentos ao longo dos anos por outros estudiosos, os quais auxiliaram na evolução do metamodelo inicial. Com isso, traz uma abordagem conceitual e considera o mundo real como algo com relacionamentos e entidades.

Um elemento essencial do Modelo Entidade Relacionamento (MER) é o Diagrama Entidade-Relacionamento, que é a representação visual de objetos de dados.

A maioria das extensões de um metamodelo tem como fundamento alguns dos métodos de abstração: agregação, classificação e generalização. Como você estudou no primeiro módulo, a abstração possibilita a separação de determinada realidade em partes, as quais serão importantes para o desenvolvimento de sistema, auxiliando na remoção da modelagem com suas peculiaridades, que não influenciam o ambiente que será modelado.

Para que você entenda o objetivo, as características principais e a sua contribuição para a construção de banco de dados do MER, o convidamos a ouvir o *podcast* adiante.



Podcast

Confira o *podcast* sobre MER.

Perdeu algum detalhe? Confira o que foi abordado no vídeo.

Olá! Vamos falar sobre o Modelo Entidade Relacionamento (MER)?

O MER auxilia na construção das estruturas de dados desenvolvidas fora de um ambiente de banco de dados com o uso de alguma linguagem de programação, podendo esta, ser a base para programação de sistemas orientados a objetos. Assim, o MER é aplicado a projetos de banco de dados, mas não na modelagem de processos de negócios.

Mas, de modo geral, somente ter o conhecimento das técnicas de modelagem não satisfaz a construção de bons modelos, pois é fundamental que se tenha a compreensão dos conceitos da realidade.

Para isso, no próximo tópico aprofunde seus estudos sobre os elementos constitutivos do MER.

Agora que você já teve o primeiro contato com o MER, é relevante nos aprofundarmos nele, sobretudo, em relação aos elementos constitutivos. Para isso, acompanhe o próximo tópico!

Elementos do MER

O MER tem três classes de objetos, sendo elas: entidades, relacionamentos e atributos.

As **entidades** se referem a qualquer coisa do mundo que se pretende guardar as informações. Elas podem ser de dois tipos, vamos conhecê-los:

Tipo físico

Como as pessoas jurídicas ou físicas, as empresas, os fornecedores, os funcionários e os clientes.

Tipo lógico

Como os objetos materiais ou abstratos, que podem ser os veículos, os produtos, os projetos, as disciplinas, os fatos e eventos, os empréstimos, as viagens ou uma venda, na maioria dos casos.

Graficamente, as entidades são simbolizadas no formato de um retângulo com o nome a qual a entidade se refere em seu centro, na maioria das vezes um substantivo no singular, como “cliente”. Além disso, elas são categorizadas em fortes, fracas e associativas. Leia abaixo para saber mais detalhadamente sobre cada uma delas:

Entidades fortes

Não dependem de outras entidades, uma vez que, mesmo sozinhas, já têm um motivo para existir. Por exemplo, a entidade “produto”, no sistema de vendas, não depende de outra para existir, pois toda a venda envolve algum produto, seja de serviço ou físico.

Entidades fracas

Dependem de outras entidades para existir, visto que, sozinhas não têm um propósito para sua existência. Por exemplo, observando a entidade “produto” de outra perspectiva, nota-se que a entidade “venda” é dependente da primeira entidade, afinal, uma venda sem produtos não apresenta sentido.

Entidades associativas

Aparecem quando necessitamos vincular uma entidade a um relacionamento existente. Importante frisar que, no modelo MER, é impossível que algum relacionamento seja diretamente associado a uma entidade.

Assim, tal relação é transformada em uma entidade associativa, possibilitando a vinculação com outras entidades. Para exemplificar, podemos usar a entidade “brinde” no contexto das entidades “produto” e “venda”. Caso se queira criar uma entidade “brinde”, esta não estaria vinculada apenas à venda, nem ao produto, mas com o item da venda que conteria a relação das duas entidades. Dessa forma, não será possível relacionar a entidade “brinde”. Logo, instancia-se a entidade associativa “item da venda”, que contém os atributos para identificar as entidades “produto” e “vendas” com os dados, como o número de série, a quantidade etc.



O **relacionamento**, por sua vez, consiste na demonstração de associações entre entidades e o mundo real ou, ainda, em um acontecimento que vincula dois objetos existentes no mundo real. A maioria das relações não possui existência conceitual ou física, enquanto outros são dependentes da associação de entidades.

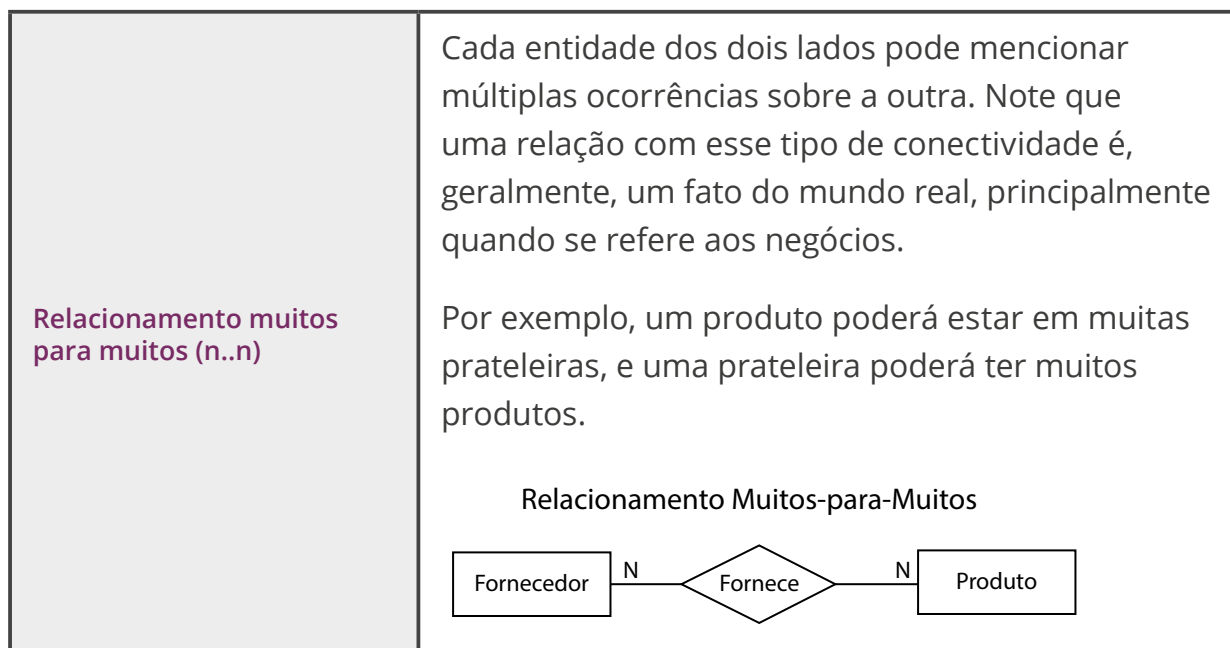
Desse modo, um relacionamento será nomeado com um verbo para conectar as duas entidades, sendo detalhado em cardinalidade, ou, para facilitar o entendimento, conectividade, que é a definição de grau de relação entre duas entidades ou tabelas, existência ou grau.

É incomum uma entidade aparecer sozinha no mundo real, independentemente de qualquer outra. Na verdade, geralmente é identificada a presença de alguma associação entre duas entidades diferentes.

Assim, a vinculação lógica entre duas ou mais entidades caracteriza um relacionamento, simbolizado pelo Diagrama Entidade-Relacionamento por meio de uma linha, a qual conecta as entidades vinculadas e apresenta um losango com o nome do relacionamento.

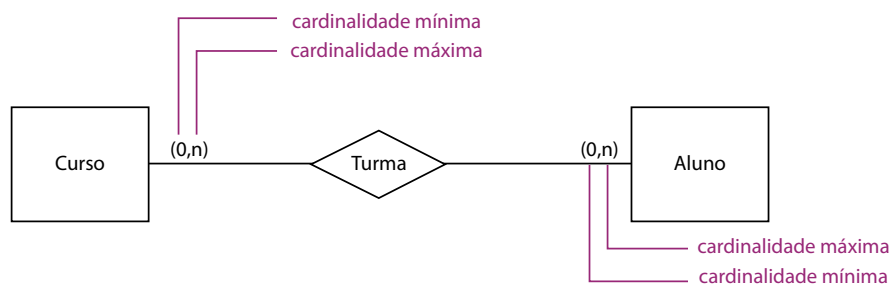
O relacionamento mais usual é o de **conectividade** ou **cardinalidade**, que pode ser classificado em três tipos distintos. Acompanhe a tabela!

<p>Relacionamento um para um (1..1)</p>	<p>Obrigatoriamente, as duas entidades se referem apenas a uma ocorrência da outra. Isto é, há uma ligação em que cada ocorrência da entidade X se vincula somente com uma ocorrência da entidade Y. Além disso, cada ocorrência da entidade Y se vincula com uma ocorrência da entidade X. Por exemplo, um computador possui somente um teclado conectado, ao passo que um teclado está conectado somente a um computador.</p> <p style="text-align: center;">Relacionamento Um-para-Um</p>  <pre> graph LR Gerente[Gerente] --- 1 --- Chefia{Chefia} Chefia --- 1 --- Seção[Seção] </pre>
<p>Relacionamento um para muitos (1..n)</p>	<p>Uma das duas entidades pode mencionar mais de uma ocorrência da outra, mas, para a outra entidade, cada uma das associações só pode se vincular a uma ocorrência da entidade. Assim, uma ocorrência da entidade X se vincula ou associa a mais de uma ocorrência da entidade Y, porém Y está vinculada somente a uma ocorrência da entidade X. Por exemplo, em um sistema de plano dental, um cliente principal pode ter mais de um dependente atrelado, mas estes estarão vinculados somente a um cliente principal.</p> <p style="text-align: center;">Relacionamento Um-para-Muitos</p>  <pre> graph LR Seção[Seção] --- 1 --- Trabalha{Trabalha} Trabalha --- N --- Funcionario[Funcionário] </pre>



#PraCegoVer: na imagem, temos um esquema indicando exemplos de cardinalidades. De cima para baixo, encontramos o relacionamento um para um (gerente (1), chefia e (1) seção), relacionamento um para muitos (seção (1), trabalha e (n) funcionário) e relacionamento muitos para muitos (fornecedor (n), fornece e (n) produto).

Além disso, a cardinalidade é categorizada em dois tipos principais, observe no exemplo abaixo:



#PraCegoVer: na imagem, temos um esquema exemplificando as cardinalidades mínima e máxima. Do lado esquerdo, encontramos um retângulo com o texto "Curso", seguido de "Turma" e "Aluno". Entre "Curso" e "Turma" há (0, n), em que "0" é a cardinalidade mínima e "n" é a cardinalidade máxima. Entre "Turma" e "Aluno" também há (0, n).

- **mínima:** determina se é obrigatório ou não o relacionamento entre duas entidades;
- **máxima:** determina o número máximo de ocorrências de uma entidade que pode fazer parte do relacionamento, devendo ser maior que zero.

Por último, mas não menos importante, os **atributos** são as peculiaridades de uma entidade que a descrevem minuciosamente, simbolizando as características essenciais de uma entidade ou de um relacionamento. Assim, cada atributo se vincula a um domínio privado, que será um grupo de valores válidos.

Os atributos são identificados durante a fase de análise de requisitos. Seu valor será determinado por uma ocorrência específica de um atributo em um relacionamento ou uma entidade. Por exemplo, uma entidade “aluno”, pode ter a identificação de matrícula, nome, curso, endereço, celular etc.

Os atributos podem ser categorizados de acordo com sua função, em três grupos principais, analise-os abaixo para saber quais são:

Atributo chave, identificador ou nominativo

Utilizado apenas para definir a identificação de alguma ocorrência de entidade, igualmente conhecido como chave primária no ambiente relacional. Quando se executa a abstração para identificação de entidades, não há chaves primárias do tipo código.

Atributos não chaves, descritores ou descritivos

São empregados para detalhar características que não são únicas, relacionadas a alguma ocorrência em particular da entidade.

Atributos referenciais

Simbolizam o vínculo de uma entidade com outra em uma relação. Necessariamente não fazem parte do objeto (entidade), mas realizam alguma espécie de referência com o outro objeto. Por exemplo, podemos mencionar que uma venda tem o RG do cliente que realiza o relacionamento com a entidade “cliente”.

Importante ressaltarmos que os atributos descritores e identificadores ainda podem ser subcategorizados quanto à sua estrutura, conforme indicado abaixo:

Simples

As características de uma entidade são determinadas por apenas um atributo, como peso, altura, nome, idade etc.

Composto

São utilizados vários atributos para determinar uma informação da entidade. Seu interior é constituído por vários elementos de dados menores, no caso, outros atributos. Os dados de um endereço podem ser compostos pelas informações de CEP, bairro, cidade, número etc. De modo geral, considera-se uma informação que contém vários dados.

Multivalorado

Seu interior é constituído por mais de um valor. Por exemplo, no caso do atributo “telefone”, uma pessoa poderá ter mais de um número telefônico para salvar no seu registro, logo, para a mesma entidade, haverá dois valores para o mesmo atributo. Nesses casos, a partir do processo de normalização, o campo terá um asterisco (*) antes do nome do atributo para indicar que é multivalorado.

Derivado

Para determinados atributos, haverá um relacionamento. Por exemplo, com base nos atributos “data-nascimento” e “idade”, para determinada pessoa podemos definir o valor de “idade atual” utilizando o atributo “data-nascimento”, em que a idade será um atributo que deriva do atributo “data-nascimento”.

Determinante

Distingue uma entidade de modo único, logo, não haverá valores repetidos, como número de matrícula, CNPJ, entre outros.

Até aqui, você estudou os três elementos constitutivos do MER, os quais são: as entidades (físicas e lógicas, que são categorizadas em fortes, fracas ou associativas), os relacionamentos (sendo a conectividade, o tipo mais utilizado, podendo ser de um para um, um para muitos e muitos para muitos) e os atributos (cujas categorias são os atributos chaves, os não chaves e os referenciais).

Contudo, a fim de que você, como desenvolvedor, possa entender o MER com mais detalhes e, assim, viabilizar os futuros projetos que envolvam esse modelo, muitas vezes, é usado o Diagrama Entidade-Relacionamento, também conhecido apenas como DER. Continue seus estudos para saber mais sobre ele.

Diagrama Entidade-Relacionamento (DER)

Conhecer os elementos do MER auxilia na concepção do modelo e, assim, contribui para a elaboração de um modelo de dados eficaz. No entanto, muitas vezes, é necessário utilizar um esquema gráfico que torne concreto o que foi projetado conceitualmente. Logo, é nesse momento que surge o Diagrama Entidade-Relacionamento (DER).

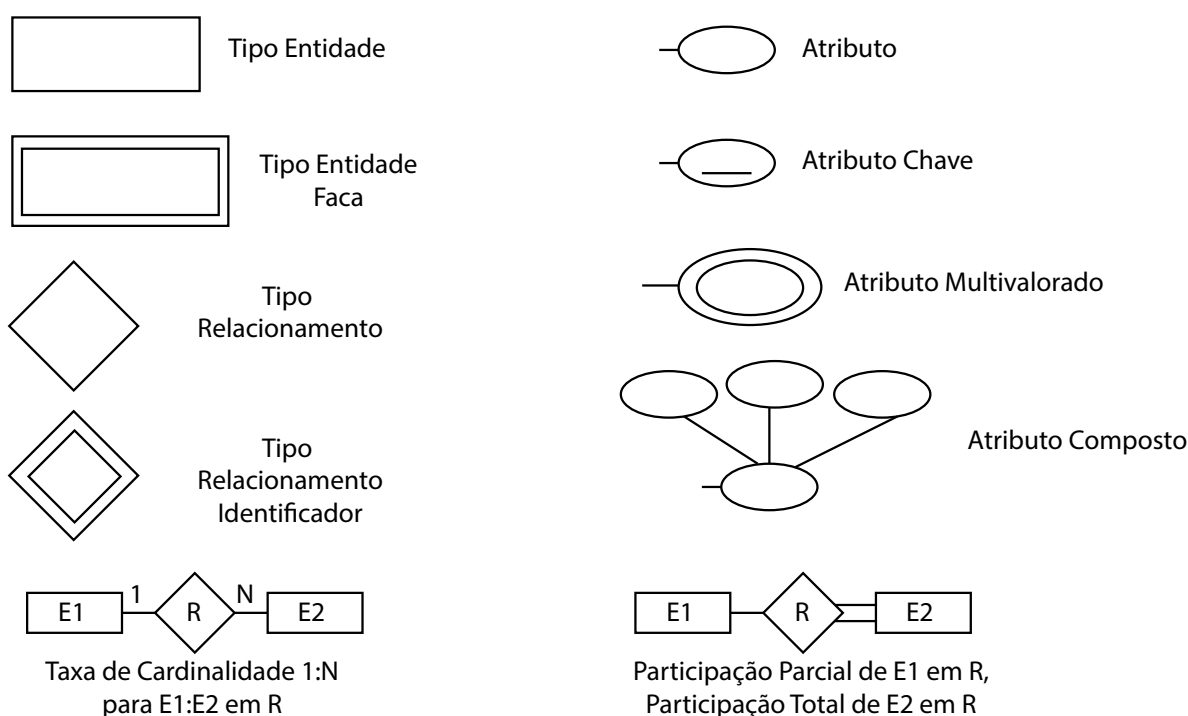
Assim, estamos falando de dois assuntos distintos: o MER se refere a um modelo conceitual, ao passo que o DER, ou diagrama ER, refere-se à simbolização gráfica do modelo (MER), sendo um tipo de fluxograma de como as entidades se relacionam.

Em algumas situações, o diagrama ER será interpretado como sinônimo do MER, sendo que este poderá ser entendido como algo abstrato para ajudar na programação do sistema, isento de uma forma de exibição das informações do MER. Assim, gerar uma representação gráfica, quando se está realizando a modelagem do domínio, seria o mais comum, com base em algumas regras.

Conforme abordamos antes, esse diagrama auxilia na comunicação da equipe de desenvolvedores, porque tal recurso possui uma linguagem compreendida pelos membros da equipe e de projetos. Desse modo, a notação original do diagrama ER será representada por retângulos (entidades), elipses (atributos) e losangos (relacionamentos).

Entretanto, a notação, recentemente, deixou de utilizar para os atributos a simbolização de elipses, passando a utilizar um modelo bem usual da UML, no qual os atributos são exibidos em listas na entidade, facilitando o entendimento.

Até aqui, você viu que os atributos podem ser categorizados, enquanto as entidades possuem cardinalidades. Dentro do DER, podemos simbolizar graficamente essas informações, conforme você pode analisar na imagem a seguir, incluindo o símbolo para cada tipo de atributo e cardinalidade. Observe com atenção!



#PraCegoVer: na imagem, temos um esquema retratando os objetos que compõem o DER. Da esquerda para a direita, de cima para baixo, temos um retângulo (trindade), um retângulo com outro dentro (entidade fraca), um losango (relacionamento), um losango com outro dentro (relacionamento identificador), um círculo (atributo), um círculo com um traço um pouco baixo do centro (atributo chave), um círculo com outro dentro (atributo multivalorado) e um círculo ligado a outros três (atributo composto). Abaixo desses objetos, temos um esquema de taxa de cardinalidade 1:N para E1:E2 em R. Do lado direito, há outro esquema, agora com participação parcial de E1 em R e participação total de E2 em R.

No Diagrama Entidade-Relacionamento, os relacionamentos são verbos, ao passo que as entidades se transformam em substantivos.

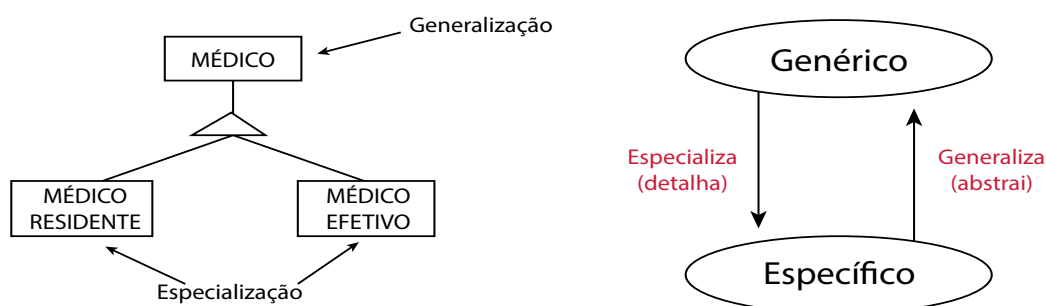


Saiba mais

Para conhecer os símbolos, as notações e maiores detalhes a respeito do diagrama ER, sugerimos a leitura do artigo [O Que é um Diagrama Entidade-Relacionamento?](#) Vale a pena conferir!

Inclusive, destacamos que o diagrama ER traz as extensões denominadas **generalização** e **especialização**, utilizadas para a representação de entidades do mundo real, as quais possuem atributos iguais, que são classificados e simbolizados em alguma hierarquia para exibir as dependências entre as entidades, que possuem uma mesma classificação.

Essas extensões possibilitam que propriedades específicas de um subgrupo das ocorrências (especializadas) sejam herdadas de uma entidade genérica ou pai. Observe a imagem a seguir para entender melhor!



#PraCegoVer: na imagem, temos dos esquemas exemplificando especialização e generalização. À esquerda, a generalização seria o médico, que pode ser dividido em médico residente e médico efetivo, que seriam as especializações. À direita, encontramos o genérico em cima e o específico embaixo, sendo que o genérico especializa (detalha) para o específico, enquanto este generaliza (abstrai) para o genérico.

Note que a **generalização** se refere a uma entidade que é um supergrupo de outra entidade, além de ser uma metodologia que generaliza vários grupos de entidades em um só grupo.

Atenção



Quando há entidades com o mesmo grupo de atributos para detalhá-las, será possível generalizá-las em uma entidade só, preservando a identificação do subconjunto a partir da inserção de algum atributo que as qualifique nas suas ocorrências.

Já a **especialização** diz respeito a uma entidade que é um subgrupo de outra, além de ser uma metodologia que categoriza o grupo de entidades em um grupo de entidades especializadas. É, portanto, a identificação de algum grupo viabilizada pela qualificação de atributos, contendo uma classe inserida em outra genérica. Como exemplo apresentado na imagem acima, temos a entidade principal “médico”, que pode ser um médico residente ou um médico efetivo.

Após estudar sobre o DER, Diagrama Entidade-Relacionamento, ou diagrama ER, você viu que ele é um fluxograma elaborado graficamente, cuja função é tornar concreto o MER.

Por consequência disso, o DER auxilia a comunicação e, por fim, otimiza o projeto de construção do modelo de banco de dados.

No entanto, para que você se aprofunde não só no MER, é relevante entender o conceito de condicionalidade de relacionamento e autorrelacionamento, o qual veremos no próximo tópico.

Condicionalidade de um relacionamento e autorrelacionamento

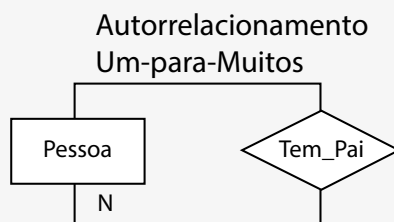
Vimos que o DER é um fluxograma baseado no Modelo Entidade Relacionamento. Mas, para que você compreenda mais sobre a aplicação deles, é necessário entender o conceito de relacionamento e de autorrelacionamento, sobretudo, associado a condicionalidade.

Um relacionamento apresentará restrições ou será condicional. Em outras palavras, é possível ter um modelo em que uma entidade se vincule a outras duas, mas pode ocorrer exclusivamente com cada uma.

Os relacionamentos reflexivos, também conhecidos como autorrelacionamentos, dão-se entre as ocorrências de uma mesma entidade. Eles apresentam dois tipos de cardinalidade: um para muitos, e muitos para muitos. Para conhecer mais sobre eles, acompanhe a seguir:

Autorrelacionamento um para muitos

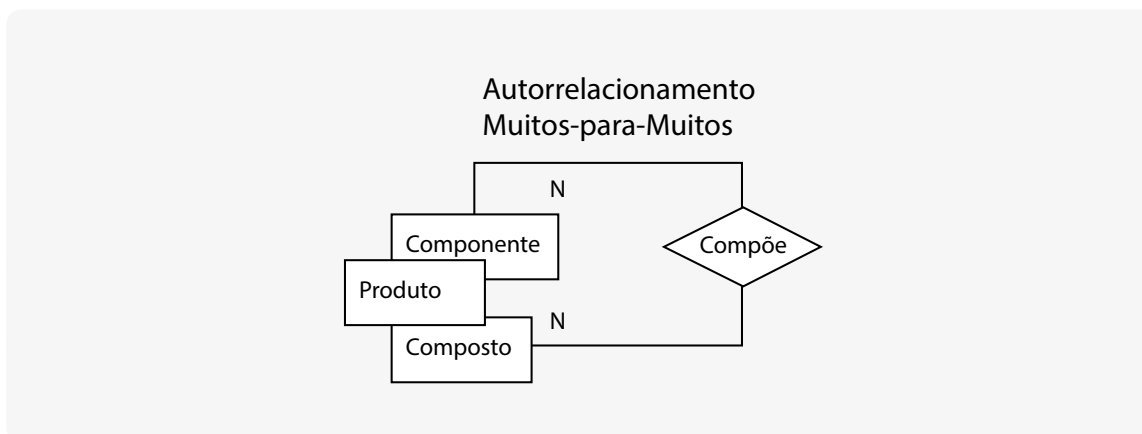
Geralmente, simboliza algum tipo de hierarquia entre as entidades. Por exemplo, a entidade “funcionários” se refere a todos os colaboradores da mesma organização, mas alguns realizam a gestão de outros funcionários, estabelecendo um autorrelacionamento de um para muitos, o qual ocorre na mesma entidade.



#PraCegoVer: na imagem, temos um esquema exemplificando autorrelacionamentos um para muito (pessoa tem pai)

Autorrelacionamento muitos para muitos

Retrata as composições de algum objeto. Por exemplo, em uma fábrica, um produto é constituído de vários outros produtos, mas um produto componente também poderá participar da composição de outros produtos.



#PraCegoVer: na imagem temos um esquema exemplificando autorrelacionamentos muitos para muitos (componente, produto e composto compõe).

Nesse sentido, foi possível observar que um relacionamento pode conter restrições, ou ser do tipo condicional. Além disso, você também notou que já o autorrelacionamento (ou ainda relacionamento reflexível) pode ser de dois tipos: um para muitos e muitos para muitos.

Desta forma, tais conceitos serão fundamentais, sobretudo, para a construção do Diagrama Entidade-Relacionamento, fator essencial para a modelagem de Banco de Dados.

Parabéns, você finalizou o módulo 2.

Aqui, você estudou sobre a história e as características do MER, Modelo Entidade Relacionamento; e sobre seus elementos, as entidades, os relacionamentos e os atributos.

Além disso, você também estudou sobre o DER, o Diagrama Entidade-Relacionamento, recurso muito importante para elaboração de modelos de bancos de dados, principalmente quando esse for elaborado por equipes. Somado a tudo isso, viu também os conceitos de condicionalidade de um relacionamento e autorrelacionamento.

No próximo módulo do curso, abordaremos um tema essencial na modelagem de dados: os conceitos necessários para manter a integridade do banco de dados!

Vamos lá?



Módulo 3

Integridade de um banco de dados e outros conceitos

Integridade de um banco de dados e outros conceitos

Durante a modelagem do banco de dados, a fase da criação da ideia até o planejamento dela é muito relevante. Contudo, não podemos deixar de lado a fase de execução dele, ou seja, da sua prática, uma vez que esse projeto precisa sair do papel e funcionar na realidade. Assim, para que isso ocorra, é necessário estabelecer a integridade de um banco de dados.

Para garantir essa integridade, Edgar Codd estabeleceu algumas regras e restrições para modelos relacionais. Dessa forma, podem ser categorizadas em: integridade referencial, de domínio, de vazio, de chave e definida pelo usuário. Assim, neste último módulo, veremos esses e outros conceitos.

Preparado(a)? Então vamos lá!

12 regras de Codd para bancos relacionais

Ao criar o Modelo Entidade Relacionamento (MER), Edgar Codd estabeleceu um conjunto de 12 regras (13, se incluirmos a zero), para definir que um banco de dados é verdadeiramente relacional. As 12 regras são caracterizadas conforme você pode ver a seguir.

Regra 0

O sistema necessita ser categorizado como um banco de dados, um sistema de gerenciamento e um sistema relacional.

Regra 1

A informação em um banco de dados relacional será apresentada em nível lógico, por valores em tabelas.

Regra 2

Os dados de um banco relacional têm a garantia de serem logicamente acessíveis, utilizando uma combinação de um nome de coluna, um nome de tabela e um valor de chave.

Regra 3

O tratamento deverá ser pelo sistema de valores nulos (ausência de dados).

Regra 4

No discriçãoário de dados, o catálogo relacional ativo tem por base o modelo relacional.

Regra 5

O SGBD relacional deve ter uma linguagem com definição, detalhamento e manipulação dos dados.

Regra 6

O tratamento deverá considerar as atualizações de visões dos dados.

Regra 7

Necessidade de haver tratamento de alto nível para atualização, inserção e eliminação de dados.

Regra 8

Os dados físicos, como a mudança na memória ou no método de acesso, devem ser independentes.

Regra 9

Os dados lógicos devem ser independentes, como nas alterações de qualquer tipo nas tabelas básicas.

Regra 10

Necessidade de haver independência das regras de integridade.

Regra 11

Indispensável ocorrência da independência de distribuição.

Regra 12

Não transposição das regras de integridade ou restrições quando se utiliza uma linguagem de baixo nível.

Essas regras foram criadas com o intuito de impossibilitar que sua visão para o banco de dados relacional fosse desfeita. Entretanto, são raros os bancos de dados que se enquadram em mais do que 10 dessas normas.

Assim, você notou que essas regras validam um banco de dados relacional, observando os principais detalhes de cada norma.

Porém, é essencial que, atrelada às regras, também sejam aplicadas as restrições de integridade para o funcionamento pleno de um banco de dados. Desta forma, vamos estudar mais sobre isso no próximo tópico.

Restrições de integridade

A integridade de dados tem por finalidade a manutenção e garantia da consistência, bem como a exatidão de dados, sendo um dos aspectos importantes na implementação e no uso dos sistemas de banco de dados. Além disso, ela será alcançada por meio da aplicação das restrições de integridade.

Em banco de dados, essas restrições podem ser categorizadas de acordo com os

cinco tipos existentes, ou seja, integridade referencial, de domínio, de vazio, de chave e definida pelo usuário. Adiante, você entenderá melhor sobre cada uma delas.



Entenda o Conceito

A **integridade referencial** garante que os dados de um banco relacional não serão corrompidos, impossibilitando que exista um registro filho sem haver um registro pai.

Dessa forma, quando um registro depende de outro, devemos aplicar regras para garantir que um registro pai não possa ser excluído se abaixo dele existirem registros filhos. Esse relacionamento é denominado por chaves estrangeiras, validadas pelo sistema antes de algum comando que possa alterar o dado.

Em resumo, a restrição de integridade referencial determina sobre os valores dos campos que aparecem em uma chave estrangeira, devendo aparecer uma chave primária da tabela referenciada. Logo, tal integridade prova a garantia de que as linhas vinculadas em duas tabelas continuarão assim, ainda que tenham sido realizadas modificações na tabela. Uma linha de determinada tabela que faz referência a outra tabela deverá referenciar a linha existente naquela tabela.



Entenda o Conceito

A **integridade do vazio**, por sua vez, determina se o preenchimento da coluna é opcional ou obrigatório, havendo a possibilidade de deixar o valor da coluna nula sem inserir algum dado.

Para entender melhor, imagine que temos zero e dado nulo, eles são opções diferentes, pois o primeiro insere o zero, enquanto o segundo não insere nada, deixando o campo em branco. A opção será aplicada de acordo com as Regras de Negócio (RN), conforme o sistema do banco de dados, sendo opcional a obrigatoriedade de preenchimento.



Conceito

Já a **integridade de chave (primária)** determina que os valores inseridos como chaves primárias serão únicos, não havendo repetição do valor, nem a chance de serem nulos.

Dessa maneira, temos que os registros serão diferentes, preservando a integridade da chave primária. A seguir, observe os dois tipos de registros da chave primária:

Tipo 1

Registros diferentes.

Tipo 2

Integridade da chave primária preservada.



Conceito

Ainda temos a **integridade definida pelo usuário**, que é a única integridade que se refere às regras de negócio, as quais são determinadas pelo usuário de banco de dados.

Em certas situações, as demais integridades podem não ser suficientes para a proteção dos dados, então o usuário tem a liberdade de criar regras que atendam às necessidades do negócio.



Conceito

Por fim, temos a **integridade de domínio**. Ela se refere ao domínio de um atributo ou um tipo do valor de um atributo que aparece em cada coluna de uma tabela, admitindo os dados do mesmo tipo. Desse modo, os valores de atributos são aceitos em cada coluna, de acordo com a informação inserida.

Por exemplo, no caso de um atributo “preço”, este somente aceitará o valor monetário, não sendo possível inserir uma data ou um texto no campo. Igualmente, em relação a um dado do tipo “data”, neste somente serão inseridos os dados do tipo data, não permitindo a inserção de texto, valor monetário ou qualquer outra informação que não seja uma data.

Essa integridade de domínio permite a inserção de regras com limitação de formato e quantidade, podendo ser categorizada de cinco formas. Vamos conhecê-las!

Restrições de *check*

Possibilita o controle dos dados armazenados em uma coluna, referente a qualquer tabela, validando uma expressão. Temos como exemplo o menor que, maior que, diferente de etc.

Nulidade

Realiza o controle se há obrigatoriedade de um valor para determinada coluna. Os valores nulos deverão ser evitados, visto que acarretam um espaço desperdiçado no banco. Assim, serão utilizados os valores nulos em situações que há valores, mas eles serão desconhecidos, ou seja, um valor nulo é conhecido, porém ausente.

Define, nas tabelas, um conjunto de atributos ou algum atributo em que a combinação ou o valor deverão ser diferentes em qualquer uma das ocorrências das tabelas.

Unique

Define que os valores de alguma coluna necessitam ser distintos (exclusivos), criando uma integridade.

Default

Define um valor padrão a ser seguido para alguma coluna.

Saiba mais



Para aprofundar seus conhecimentos a respeito do tema modelagem de dados, indicamos que entre no site [DevMedia](#). O portal certamente será de grande auxílio para os seus conhecimentos! Vale a pena conferir!

Assim, neste último tópico, você compreendeu mais acerca do aspecto funcional da modelagem de banco de dados, o conceito de integridade. A partir disso, você estudou sobre as 12 regras de Codd para banco de dados relacional e o conceito de restrição de integridade e seus diferentes tipos.

Fechamento

Parabéns! Você finalizou o curso de Modelagem de Dados.

Na contemporaneidade, em que cada vez mais novos softwares são criados para diferentes interfaces e para as diversas finalidades, desde as mais simples até as mais complexas, é imprescindível que os dados e informações ligadas ao programa, a ser desenvolvido, sejam organizados e garantam o pleno funcionamento dele.

Diante desse grande desafio, a Modelagem de Dados surge como uma grande aliada. E, considerando esse cenário, este curso ofereceu não só a base necessária para entender a definição e importância dela, mas também evidenciou a sua aplicabilidade, quais os passos essenciais para construir um banco de dados eficiente, como proteger os dados e manter sua integridade.

Esperamos que tenha conseguido compreender as particularidades envolvidas no assunto, descobrindo novos conceitos ou relembrando o básico para construir um banco de dados adequado, seguro e íntegro.

Até a próxima!

Referências

BANCOS de dados - **Projeto de banco de dados**. [S. l.], [s. d.]. 1 vídeo (37 min). Publicado pelo canal Ramon Souza. Disponível em: <https://www.youtube.com/watch?v=62jiMNUmGF8>. Acesso em: 20 maio 2021.

BONIM, N. Chave primária e chave estrangeira: entenda a diferença entre elas. **O Manual do Freelancer**, [s. l.], 14 jun. 2019. Disponível em: <https://omanual dofreeslancer.com/diferenca-chave-primaria-e-chave-estrangeira/>. Acesso em: 21 fev. 2021.

BRANDÃO, R. O que é um banco de dados relacional? **INSiders Business Data**, [s. l.], 9 nov. 2017. Disponível em: <https://www.datainsiders.com.br/post/o-que-e-um-banco-de-dados-relacional>. Acesso em: 21 fev. 2021.

CONCEITOS fundamentais de banco de dados. **Devmedia**, [s. l.], [s. d.]. Disponível em: <https://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649>. Acesso em: 21 fev. 2021.

DATE, C. **Introdução a sistemas de bancos de dados**. 8. ed. Amsterdã: Elsevier, 2004.

DEVMEDIA. [S. l.], [s. d.]. Disponível em: <https://www.devmedia.com.br/>. Acesso em: 8 maio 2021.

DIAGRAMA Entidade-Relacionamento - DER. **Banco de Dados I**, [s. l.], [s. d.]. Disponível em: <https://sites.google.com/site/fkbancodedados1/modelodados/der>. Acesso em: 21 fev. 2021.

DORNELLES, C. A. Generalização e especialização. **CAD COBOL**, [s. l.], [s. d.]. Disponível em: http://www.cadcobol.com.br/db2_novo_generalizacao_e_especializacao.htm. Acesso em: 21 fev. 2021.

FAIS, R. M. Elementos básicos de um banco de dados. **RMFAIS**, [s. l.], [s. d.]. Disponível em: http://www.rmfaiss.com/rmfaiss/artigos/table.php?_codigo=80. Acesso em: 21 fev. 2021.

FRANCESCHINI, L. Técnicas de sistema e processamento de dados I. **Professor Rodrigo**, [s. l.], [s. d.]. Disponível em: <https://profrodrigo.leandrofranceschini.com>.

[br/tecnicas-de-sistema-e-processamento-de-dados-i/](#). Acesso em: 21 fev. 2021.

FREEPIK. **Recursos gráficos para todos**. Freepik, 2021. Disponível em: <https://br.freepik.com/>. Acesso em: 28 de out. de 2021.

GUERRA, B. O que é integridade de dados e por que ela é importante? **Inteligência de Negócios**, [s. l.], 6 jan. 2020. Disponível em: <https://blog.in1.com.br/o-que-%C3%A9-integridade-de-dados-e-por-que-ela-%C3%A9-importante>. Acesso em: 21 fev. 2021.

HEUSER, C. A. **Banco de dados relacional**: conceitos, SQL e administração. Joinville: Clube de Autores, 2019.

_____. **Projeto de banco de dados**. 6. ed. Porto Alegre: Bookman, 2017. v. 4.

MACHADO, F. **Banco de dados**: projeto e implementação. 3. ed. São Paulo: Saraiva, 2018.

_____. **Banco de dados**: projetos e implementação. 4. ed. São Paulo: Érica, 2020.

MACHADO, F. N. R.; ABREU, M. P. de. **Projeto de banco de dados**: uma visão prática. 17. ed. São Paulo: Saraiva Educação S.A., 2018. v. 3.

MODELAGEM de dados - Modelo conceitual, lógico e físico. [S. l.], [s. d.]. 1 vídeo (23 min). Publicado pelo canal Miriam TechCod. Disponível em: <https://www.youtube.com/watch?v=8CkMX2qXgdY>. Acesso em: 20 maio 2021.

MACÊDO, D. Modelagem relacional – Teoria. **Diego Macêdo**: Um Pouco de Tudo sobre TI, [s. l.], 16 ago. 2012. Disponível em: <https://www.diegomacedo.com.br/modelagem-relacional-teoria/>. Acesso em: 21 fev. 2021.

MODELAGEM de dados: modelo conceitual, modelo lógico e físico. **Professor Digital**, [s. l.], [s. d.]. Disponível em: <https://www.luis.blog.br/modelagem-de-dados-modelo-conceitual-modelo-logico-e-fisico.html>. Acesso em: 21 fev. 2021.

MODELO Entidade Relacionamento (MER) e Diagrama Entidade-Relacionamento (DER). **Devmedia**, [s. l.], [s. d.]. Disponível em: <https://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332>. Acesso em: 21 fev. 2021.

NORMALIZAÇÃO. **Banco de Dados I**, [s. l.], [s. d.]. Disponível em: <https://sites.google.com/site/fkbancodedados1/normalizacao>. Acesso em: 21 fev. 2021.

NORMALIZAÇÃO de dados e as formas normais. **Professor Digital**, [s. /], [s. d.]. Disponível em: <https://www.luis.blog.br/normalizacao-de-dados-e-as-formas-normais.html>. Acesso em: 21 fev. 2021.

O QUE é um Diagrama Entidade-Relacionamento? **Lucidchart**, [s. /], [s. d.]. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-entidade-relacionamento/>. Acesso em: 21 fev. 2021.

REBELO, I. 3 modelos conceituais. **IRLABR**. [s. /], 2011. Disponível em: <https://irlabr.wordpress.com/apostila-de-ihc/parte-1-ihc-na-pratica/modelos-conceituais/>. Acesso em: 21 fev. 2021.

SHUTTERSTOCK. **Transforme ideias em conquistas**. Shutterstock, 2021. Disponível em: <https://www.shutterstock.com/pt/>. Acesso em: 28 de out. de 2021.

SIQUEIRA, F. de. Projeto de banco de dados. **Google Sites**, [s. /], [s. d.]. Disponível em: <https://sites.google.com/site/uniplibancodedados1/aulas/aula-3---projeto-de-banco-de-dados>. Acesso em: 20 maio 2021.

SOUZA, I. de. Banco de dados: saiba o que é, os tipos e a importância para o site da sua empresa. **Rock Content**, [s. /], 25 fev. 2020. Disponível em: <https://rockcontent.com/br/blog/banco-de-dados/>. Acesso em: 21 fev. 2021.

 fundação **bradesco** | escola virtual