



Colégio ULBRA São Lucas

27ª Coordenadoria Regional de Educação Deliberação nº 451/2016 CEE/RS
ASSOCIAÇÃO EDUCACIONAL LUTERANA DO BRASIL



APOSTILA 5

BANCO DE DADOS



CRÉDITOS:

Autora: Camille Ferreira | Capa: Camille e Gabriel Sandes | Revisão técnica: Prof. Jeferson Faleiro Leon e Prof. Éder Oliveira de Rosso. [Curso de Banco de Dados] [VC Ensinos]



SUMÁRIO

PROCESSO DE BANCO DE DADOS	3
1. ETAPAS DO PROCESSO DE DESENVOLVIMENTO	4
ENTIDADES, RELACIONAMENTOS E ATRIBUTOS	6
1. ENTIDADES	6
2. RELACIONAMENTOS	7
3. ATRIBUTOS	8
4. EXEMPLO PRÁTICO	9
NOTAÇÃO DE PETER CHEN (MODELO ER)	10
1. EXEMPLO DE NOTAÇÃO	11
2. VANTAGENS DE NOTAÇÃO DE PETER CHEN	11
TIPOS DE DADOS	12
CARDINALIDADE	14
1. NOTAÇÃO PARA CARDINALIDADE NO MODELO ER	15
2. IMPORTÂNCIA	15
3. EXEMPLOS DE CARDINALIDADE	15



PROCESSO DE DESENVOLVIMENTO DE BANCO DE DADOS

O processo de desenvolvimento de banco de dados envolve um conjunto de etapas estruturadas que visam projetar, implementar e manter um sistema de banco de dados que atenda às necessidades de uma organização. Esse processo é fundamental para garantir que os dados sejam armazenados de maneira eficiente, segura e consistente.

O processo de desenvolvimento de banco de dados é composto por etapas estruturadas que visam projetar, implementar e manter um sistema eficiente e confiável. Ele começa com o levantamento de requisitos, onde são identificadas as informações necessárias e os objetivos do sistema.

Segue-se a modelagem conceitual, que cria uma representação abstrata dos dados usando o modelo Entidade-Relacionamento (ER), focando nas entidades, atributos e relacionamentos. Depois, na modelagem lógica, esse modelo é transformado em tabelas e chaves primárias, aplicando normalização para garantir consistência e eliminar redundâncias.

Na modelagem física, o esquema lógico é adaptado ao ambiente do SGBD, definindo índices e estruturas para otimizar o desempenho. Após isso, vem a implementação, onde o banco é criado e os dados são inseridos.

A fase de testagem é válida para consultas, regras de integridade e desempenho. Por fim, na manutenção, ajustes e atualizações são realizados para atender a novas demandas e melhorar a eficiência, garantindo que o sistema continue alinhado às necessidades da organização.



1. Etapas do Processo de Desenvolvimento:

Definição de Requisitos:

- Coletar e documentar as necessidades dos usuários e da organização. • Identificar quais dados serão armazenados, como serão usados e as restrições associadas.

Modelagem Conceitual:

- Criar um modelo conceitual que descreve os dados e seus relacionamentos, geralmente usando diagramas como o Modelo Entidade-Relacionamento (ER).

Modelagem Lógica:

- Converter o modelo conceitual para um formato compatível com o modelo de dados que será usado no banco (como o modelo relacional).
- Definir tabelas, atributos, chaves primárias e estrangeiras.

Modelagem Física:

- Traduzir o modelo lógico para a implementação específica no Sistema de Gerenciamento de Banco de Dados (SGBD) escolhido.
- Considerar aspectos como índices, particionamento e estrutura de armazenamento.

Implementação:

- Criar o banco de dados no SGBD com base no modelo físico.
- Carregar os dados iniciais e configurar permissões de acesso.



Teste:

- Validar a funcionalidade do banco de dados, como consultas, inserções, atualizações e exclusões.
- Verificar a conformidade com os requisitos de desempenho e segurança.

Manutenção e Monitoramento:

- Atualizar o banco de dados conforme as necessidades mudam.
- Monitorar desempenho, integridade dos dados e segurança.

Otimização:

- Realizar ajustes para melhorar o desempenho do banco de dados, como criar índices ou refatorar consultas.

Ferramentas e Técnicas Usadas:

- Ferramentas de modelagem: MySQL Workbench, ER/Studio, DBDesigner.
- SGBDs: MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server.
- Técnicas de normalização: Para reduzir redundâncias e inconsistências nos dados.



ENTIDADES, RELACIONAMENTOS E ATRIBUTOS

No contexto de bancos de dados e modelagem conceitual, entidades, relacionamentos e atributos são elementos fundamentais que ajudam a organizar e representar os dados de maneira lógica e estruturada.

1. Entidades

Uma entidade é algo no mundo real que pode ser identificado e armazenado no banco de dados. Geralmente, representa um objeto, pessoa, lugar, evento ou conceito sobre o qual informações precisam ser guardadas.

Exemplos de entidades:

- **Pessoa** (com atributos como nome, idade e CPF).
- **Produto** (com atributos como código, descrição e preço).
- **Carro** (com atributos como placa, modelo e ano).

No modelo entidade-relacionamento (ER), as entidades são representadas por retângulos.



2. Relacionamentos

Um relacionamento descreve a associação entre duas ou mais entidades. Ele define como as entidades interagem ou se conectam entre si.

Tipos de relacionamentos:

- 1:1 (Um para Um): Uma pessoa tem um único passaporte.
- 1:N (Um para Muitos): Um cliente pode fazer vários pedidos, mas cada pedido pertence a um único cliente.
- N:N (Muitos para Muitos): Um aluno pode estar matriculado em várias disciplinas, e uma disciplina pode ter vários alunos.

No Modelo ER, os relacionamentos são representados por losangos, conectados às entidades envolvidas.

Exemplo:

- Entidades: Cliente e Pedido.
- Relacionamento: "Faz" (um cliente faz vários pedidos).

Tipos de relacionamentos:

- 1:1 (Um para Um): Uma pessoa tem um único passaporte.
- 1:N (Um para Muitos): Um cliente pode fazer vários pedidos, mas cada pedido pertence a um único cliente.
- N:N (Muitos para Muitos): Um aluno pode estar matriculado em várias disciplinas, e uma disciplina pode ter vários alunos.

No Modelo ER, os relacionamentos são representados por losangos, conectados às entidades envolvidas.

Exemplo:

- Entidades: Cliente e Pedido.
- Relacionamento: "Faz" (um cliente faz vários pedidos).



3. Atributos

Os atributos são as propriedades ou características que descrevem uma entidade ou um relacionamento. Cada atributo contém valores que fornecem informações sobre as instâncias.

Tipos de atributos:

- **Simple:** Valores indivisíveis, como idade ou nome.
- **Compostos:** Podem ser divididos em partes menores, como endereço (composto por rua, cidade, CEP).
- **Derivados:** Calculados a partir de outros atributos, como idade (derivada da data de nascimento).
- **Chave primária:** Um atributo ou conjunto de atributos que identifica unicamente uma entidade.

Exemplos:

- Para a entidade Pessoa, os atributos podem ser: nome, CPF e data de nascimento.
- Para o relacionamento "Matrícula" entre Aluno e Disciplina, um atributo pode ser data de matrícula.

No Modelo ER, os atributos são representados por elipses conectadas às suas respectivas entidades ou relacionamentos.



4. Exemplo Prático:

Um sistema de biblioteca:

Entidades:

- **Livro:** Representa os livros do acervo.
- **Usuário:** Representa as pessoas que pegam livros emprestados.
- **Empréstimo:** Representa a ação de pegar livros.
- **Relacionamento:**
- **"Realiza":** Relaciona o **Usuário** ao **Empréstimo**.
- **"Inclui":** Relaciona o **Empréstimo** ao **Livro**.

Atributos:

- **Livro:** Título, autor, ISBN.
- **Usuário:** Nome, CPF, endereço.
- **Empréstimo:** Data de empréstimo, data de devolução.

Esse modelo ajuda a entender como os dados se conectam e facilita o desenvolvimento e gerenciamento de um banco de dados eficiente.



NOTAÇÃO DE PETER CHEN (MODELO ER)

A notação de Peter Chen é uma abordagem gráfica usada para representar o Modelo Entidade-Relacionamento (ER), desenvolvida por Peter Pin-Shan Chen em 1976. Essa notação é amplamente utilizada em modelagem conceitual de bancos de dados para descrever as estruturas de dados e suas relações de forma clara e intuitiva.

Principais Elementos da Notação de Peter Chen:

Entidades:

- Representadas por retângulos.
- Entidade forte: Existe independentemente de outras entidades.
- Entidade fraca: Depende de uma entidade forte e é identificada por uma chave estrangeira.

Atributos:

- Representados por elipses conectadas às entidades ou relacionamentos.

Tipos de atributos:

- **Simple:** Atributos indivisíveis (ex.: Nome, Idade).
- **Compostos:** Atributos que podem ser subdivididos (ex.: Endereço dividido em Rua, Cidade, Estado).
- **Multivalorados:** Podem ter mais de um valor (ex.: Telefones).
- **Derivados:** Calculados a partir de outros atributos (ex.: Idade derivada da Data de Nascimento).
- **Chave Primária:** Um atributo que identifica unicamente cada instância de uma entidade.



Relacionamentos:

- Representados por losangos conectados às entidades participantes.

Incluem:

- Cardinalidade: Define a quantidade de instâncias de uma entidade que podem estar associadas a outra (1:1, 1:N, N:M).
- Atributos de Relacionamento: Propriedades associadas ao relacionamento (ex.: Data da Matrícula no relacionamento entre Aluno e Curso).

Conexões:

- As conexões entre entidades e relacionamentos são feitas com linhas.
- As linhas podem incluir anotações sobre participação total (linha dupla) ou participação parcial (linha simples).

1. Exemplo de Notação:

Imagine um sistema acadêmico:

- Entidades: Aluno e Curso.
- Relacionamento: "Matrícula", que conecta Aluno e Curso.
- Atributos:
 - Aluno: Matrícula (chave primária), Nome, Data de Nascimento.
 - Curso: Código do Curso (chave primária), Nome do Curso.
 - Matrícula: Data de Matrícula (atributo do relacionamento).

Esse modelo seria representado com retângulos (Aluno, Curso), losango (Matrícula) e elipses conectadas para cada atributo.

2. Vantagens da Notação de Peter Chen:

- **Clareza visual:** Facilita o entendimento mesmo para usuários não técnicos.
- **Versatilidade:** Pode ser aplicada a diferentes domínios de problemas.
- **Padronização:** Amplamente reconhecida e usada em ferramentas



TIPOS DE DADOS

Tipos de Dados em bancos de dados são categorias de valores que podem ser armazenados nas tabelas, representando informações de diferentes formas. A definição do tipo de dado de cada campo ou atributo é crucial, pois determina como os dados são armazenados, manipulados e acessados. Os principais **tipos de dados** usados em bancos de dados, especialmente em sistemas de gerenciamento como MySQL, PostgreSQL, Oracle e outros:

Tipos Numéricos:

- **Inteiros (INT, INTEGER, SMALLINT, BIGINT):** Armazenam números inteiros. A escolha depende do tamanho do número.
- **Ponto flutuante (FLOAT, DOUBLE, REAL):** Armazenam números com casas decimais. São úteis para cálculos precisos com valores não inteiros.

Tipos de Texto:

- **Caractere (String) (CHAR, VARCHAR):** Usados para armazenar sequências de caracteres.
- **CHAR:** Tamanho fixo. Se o valor inserido for menor que o tamanho definido, é preenchido com espaços.
- **VARCHAR:** Tamanho variável. Não preencher com espaços extras.
- **Texto Longo (TEXT, CLOB):** Para armazenar grandes volumes de texto.

Tipos de Data e Hora:

- **Data (DATE):** Armazena apenas a data (ano, mês, dia).
- **Hora (TIME):** Armazena apenas a hora (horas, minutos, segundos).
- **Data e Hora (DATETIME, TIMESTAMP):** Armazenam tanto a data quanto a hora (ano, mês, dia, horas, minutos, segundos).
- **Intervalo de Tempo (INTERVAL):** Representa um período de tempo entre duas datas/horas.



Tipos Lógicos (Booleanos):

- **Booleano** (BOOLEAN): Armazena valores TRUE ou FALSE. Em alguns sistemas, pode ser representado como 0 (falso) ou 1 (verdadeiro).

Tipos Binários:

- **Binário** (BLOB, BINARY, VARBINARY): Usados para armazenar dados binários, como imagens, vídeos e arquivos.
- **BLOB** (Binary Large Object): Para grandes quantidades de dados binários.

Tipos Geoespaciais:

- **Geometria** (GEOMETRY, POINT, LINESTRING, POLYGON): Usados para armazenar dados espaciais e geográficos, como coordenadas geográficas, mapas, polígonos e caminhos.

Outros Tipos:

- **Identificador Único** (UUID): Usado para armazenar identificadores únicos universais.
- **Enumerado** (ENUM): Permite armazenar um conjunto limitado de valores possíveis, como "Pequeno", "Médio" ou "Grande".
- **JSON** (JSON, JSONB): Armazena dados no formato de objetos JSON, útil para armazenar dados semi-estruturados.



CARDINALIDADE

Se refere à quantidade de instâncias de uma entidade que podem ou devem estar associadas a uma instância de outra entidade dentro de um relacionamento. Ela descreve a natureza e o limite das interações entre as entidades. A cardinalidade é um conceito fundamental no Modelo Entidade-Relacionamento (ER), pois define como as entidades se relacionam entre si, ajudando a estruturar a base de dados de forma eficiente.

Cardinalidade 1:1 (Um para Um):

- Cada instância de uma entidade A está associada a no máximo uma instância de uma entidade B, e cada instância de B está associada a no máximo uma instância de A.
- Exemplo: Uma pessoa tem um único passaporte, e cada passaporte pertence a uma única pessoa.

Cardinalidade 1:N (Um para Muitos):

- Cada instância de uma entidade A pode estar associada a várias instâncias da entidade B, mas cada instância de B está associada a no máximo uma instância de A.
- Exemplo: Um autor pode escrever vários livros, mas cada livro tem apenas um autor.

Cardinalidade N:N (Muitos para Muitos):

- Cada instância de A pode estar associada a várias instâncias de B, e cada instância de B pode estar associada a várias instâncias de A.
- Exemplo: Um aluno pode estar matriculado em várias disciplinas, e uma disciplina pode ter vários alunos.



1. Notação para Cardinalidade no Modelo ER:

As cardinalidades geralmente são representadas nas linhas de relacionamento.

Exemplo:

- No relacionamento 1:N, o lado de "muitos" (N) é indicado com um símbolo de multiplicidade (como um "coração" ou linha simples) e o lado "um" com uma única linha.
- No relacionamento N:M, são usados dois símbolos de multiplicidade, representando que qualquer instância de uma entidade pode estar associada a várias instâncias da outra entidade.

2. Importância:

A definição correta da cardinalidade é essencial para o design de banco de dados, pois:

- Garante que as relações entre as entidades sejam representadas de forma fiel.
- Ajuda a evitar problemas como redundância de dados ou inconsistências.
- Facilita o processo de normalização, uma técnica de organização dos dados para evitar redundâncias.

3. Exemplos de Cardinalidade:

- **1:1**: Cada funcionário tem um único cartão de identidade.
- **1:N**: Uma empresa tem vários departamentos, mas cada departamento pertence a apenas uma empresa.
- **N:N**: Um estudante pode se inscrever em vários cursos, e um curso pode ter vários estudantes.



REFERÊNCIAS BIBLIOGRÁFICAS

HEUSER, Carlos Alberto. *Projeto de Banco de Dados*. 6. ed. Porto Alegre: Bookman, 2009.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. *Sistemas de Banco de Dados*. 6. ed. São Paulo: Pearson, 2011.

DATE, C. J. *Introdução a Sistemas de Banco de Dados*. 8. ed. São Paulo: Pearson Addison Wesley, 2004.

Artigos Científicos:

HERNANDEZ, M. A.; KAMBHAMPATI, S. *Integration of heterogeneous databases: A survey*. ACM Computing Surveys, v. 33, n. 3, p. 211–271, 2001.

GUERROUAOUI, R.; KAPRINSKI, P. *Relational database transactions: concepts and techniques*. Journal of Database Management, v. 28, n. 4, p. 15-32, 2022.

Capítulos de Livros:

FERREIRA, Ana Lúcia. Modelagem conceitual de dados. In: SOUSA, Carlos (Org.). *Fundamentos de Banco de Dados*. São Paulo: Atlas, 2015. p. 45-78.

DATE, C. J. Modelagem relacional. In: DATE, C. J. *Introdução a Sistemas de Banco de Dados*. 8. ed. São Paulo: Pearson Addison Wesley, 2004. p. 200-220.

Trabalhos Acadêmicos (TCC, Dissertações e Teses):

SANTOS, Fernanda A. *Um estudo sobre modelos de dados: da hierarquia ao modelo relacional*. 2021. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de São Paulo, São Paulo, 2021.

Documentos Eletrônicos:

GARCIA, Carlos. Modelagem de Dados: conceitos fundamentais. Disponível em: <https://www.modelagemdedados.com.br>. Acesso em: 04 dez. 2024.