

# Modelagem prática para banco de dados

- Análise de domínio
- Técnica de análise por semântica
- Diagramação entidade-relacionamento
- Elementos de banco de dados

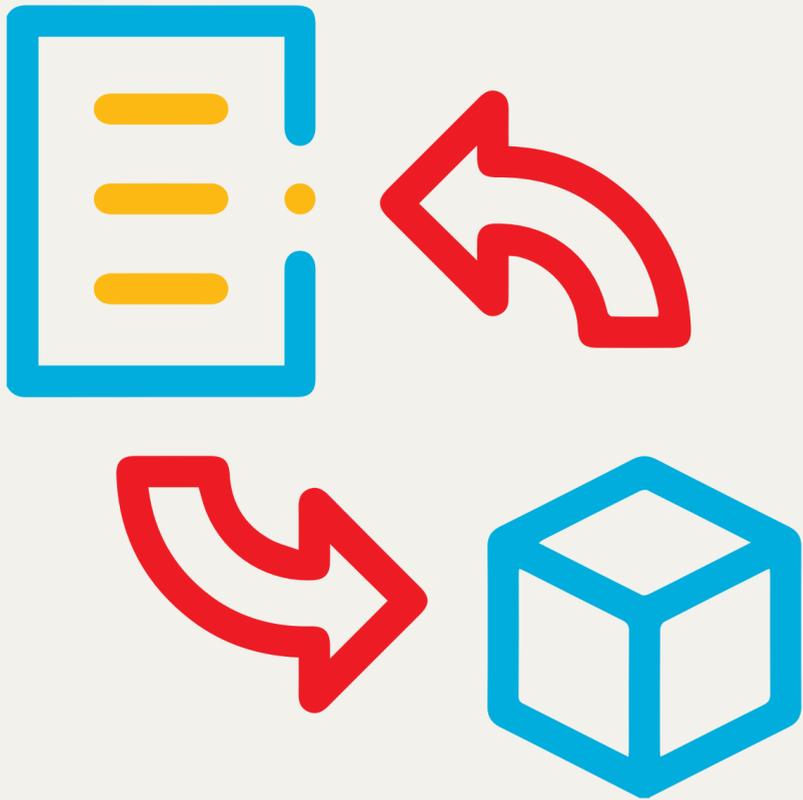
Aprenda a modelar banco de dados para desenvolver aplicativos



# SUMÁRIO



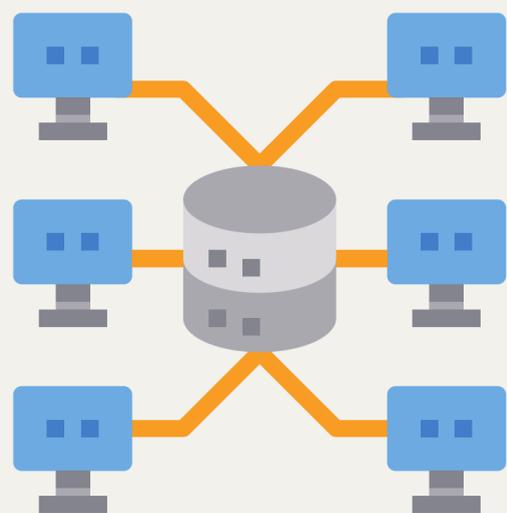
Introdução	4
Banco de Dados	5
Capítulo 01 - Modelagem para Banco de dados	5
Modelagem para BD	6
O processo da modelagem semântica	7
Análise de domínio e descrição textual	8
Elementos da modelagem de BD	9
Atributos e relacionamentos	10
Capítulo 2 - Modelagens conceitual e lógica	11
Descritivo de domínio para extrair elementos de BD	12
Elementos de BD segunda a técnica semântica	13
Capítulo 3 - Diagrama conceitual e lógico do BD	16
Capítulo 4 - Normalização	20



## Sobre o autor

# Sergio Fred Andrade

PhD in Science, Master in Systems and Computing, specialist in Advanced Computing and Distance Education, Bachelor in Information Systems and Mathematics. He has experience in computing, analysis and information systems, database, data science, and software engineering.



# INTRODUÇÃO

Neste e-book vamos abordar a modelagem para banco de dados relacional, procedimentos e processos para projeto de sistemas que usam banco de dados para armazenar e acessar informações.

A modelagem para banco de dados é realizada em três etapas: modelo conceitual, o modelo lógico e o modelo físico.

O modelo conceitual é a fase inicial onde o projetista retrata a realidade do domínio que ele quer virtualizar no banco de dados. É o retrato do mundo real com seus objetos, associações, respeitando suas regras e restrições, exatamente como acontece na realidade dos negócios, nos sistemas de produção, nas operações entre pessoas ou empresas.

O modelo lógico é a representação do modelo conceitual na linguagem técnica relacional para reconhecimento das máquinas e a preparação para o modelo físico que é a implementação em linguagem, SQL para permitir o armazenamento e recuperação das informações.

Iremos começar a trabalhar com a modelagem conceitual e os elementos importantes para banco de dados, como as entidades, seus atributos e suas relações, exatamente como acontecem com os objetos do mundo real que se pretende retratar no banco de dados.

# CAPÍTULO 1 - Modelagem para Banco de Dados

Banco de dados (BD) relacional é uma coleção de dados estruturados em tabelas que serve para armazenar e recuperar muitos dados de determinados domínios ou mini mundos reais. Geralmente são aplicados em memória externa, em discos rígidos (HD) ou memória flash (SSD) e são tratado por Sistemas Gerenciadores de Banco de Dados (SGBD).

Domínios são assuntos, processos ou sistemas organizacionais (vendas, estoques, fluxo de caixa, folha de pagamento, gestão de passagem aérea, e outros) que são fechados com suas características, mas podem ser compartilhados e integrados facilmente.

SGBD é um ambiente de desenvolvimento e aplicação de BD, formado por um conjunto de software que servem para cumprir a missão do BD como modelagem, armazenamento, edição, recuperação de informação, segurança, tratamento de concorrência em acessos e outros.

Neste livro iremos abordar banco de dados não só pela tecnologia relacional mas também numa visão mais moderna como o paradigma da orientação a objetos para tratar os elementos de BD, como entidades, atributos, relacionamentos, cardinalidades e chaves.

Um banco de dados geralmente é usado em processos organizacionais e implementado para utilização em sistemas de informação ou aplicativos, nas diversas plataformas conhecidas como desktop, web, mobile.

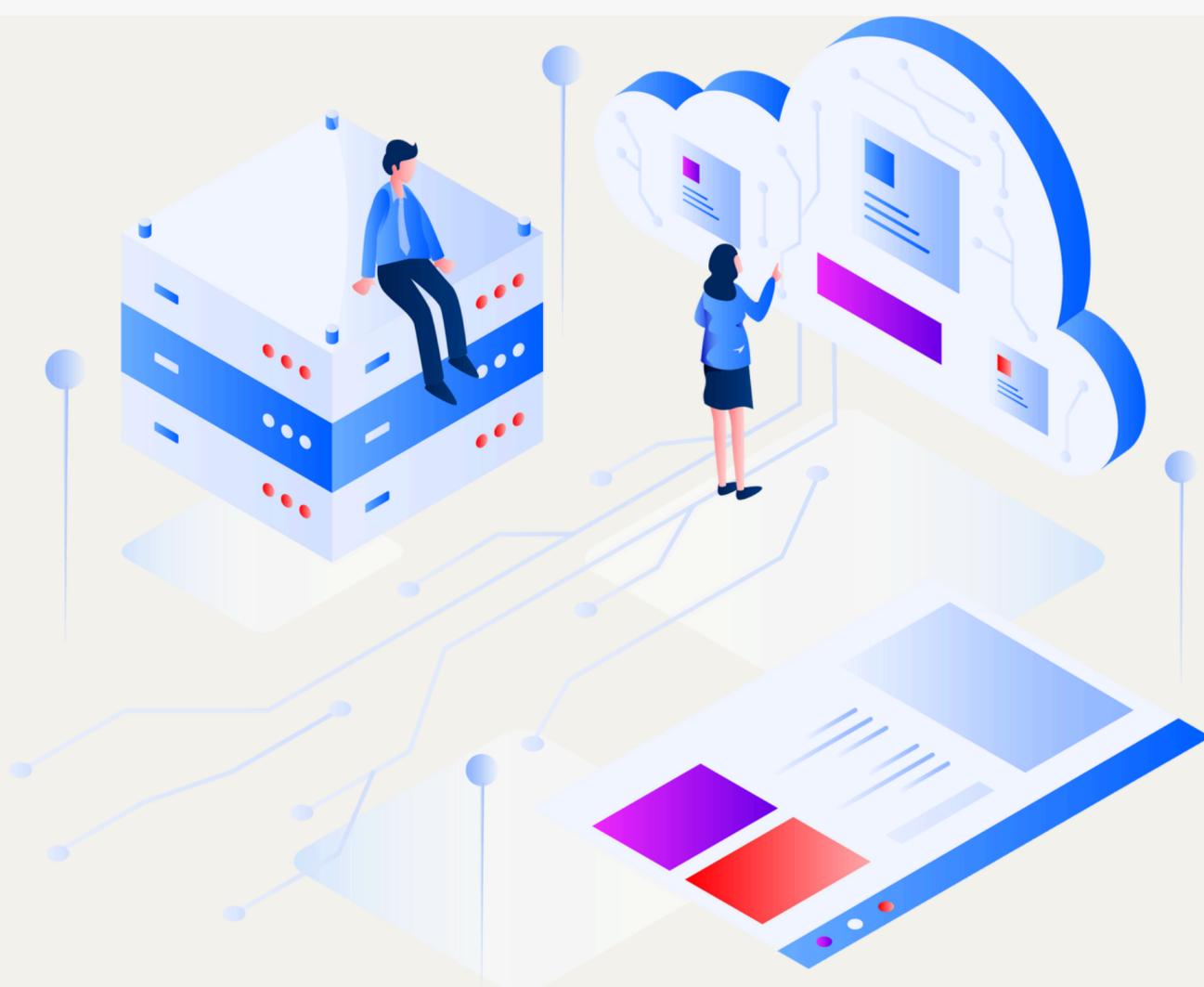
# Modelagem de Banco de dados

---

A modelagem para BD inicia com o modelo conceitual, que é uma representação abstrata dos dados de um sistema de informação. Ele descreve os tipos de informações a serem armazenadas nas tabelas, as relações entre elas e as associações presentes nos dados, sem se preocupar com detalhes de implementação.

Este livro visa fornecer uma introdução completa ao modelo conceitual, explorando seus elementos principais, desde a definição de entidades e atributos até a normalização e validação do modelo lógico de dados.

Da modelagem conceitual irá resultar na modelagem lógica com elementos de BD respeitando a sintaxe, ou seja, as regras para implementar fisicamente o projeto do BD. Isso aplicará as entidades com chaves identificadoras, os atributos com tipos de dados e tamanho e relacionamentos com cardinalidades



# O processo de modelagem conceitual pela técnica da semântica

---

A modelagem conceitual pela técnica da semântica é uma abstração do domínio que será retratado pelo banco de dados, posto num descritivo e analisado segundo os termos ou palavras presentes nesse texto.

01

## **Análise do domínio e sua descrição textual**

É a análise dos requisitos do sistema que acessará o BD. Serão descritos num texto o processo, os serviços, os objetos, seus atributos e suas associações. Deve-se respeitar as regras do negócio e como elas serão apresentadas.

02

## **Análise semântica para extrair entidades e relacionamentos**

Do texto descritivo do domínio devem ser vistos os substantivos como candidatos à entidades seus atributos, dos verbos ou adjetivos os relacionamentos e suas cardinalidades.

03

## **Definições dos atributos e relacionamentos**

Os atributos são propriedades das entidades e os representam. Devem ser únicos e sem repetições, quando for de uma entidade não será de outras. São identificadores ou descritivos.

04

## **Relacionamentos e cardinalidades entre ocorrências das entidades**

Relacionamentos são associações entre entidades e expressam ocorrências quantitativas ou cardinalidades, como uma pessoa para um cpf ou um pai para muitos filhos.

## Análise do domínio e sua descrição textual

---

Trata-se do conhecimento sobre os processos organizacionais que se pretende projetar no banco de dados. Importante saber:

1. As técnicas usadas para conhecer as atividades no domínio são aplicadas com questionários, entrevistas, reuniões para debates com os usuários, observação de cenários, análises de documentos.
2. Após coletadas as informações prepara-se um descritivo das atividades básicas e importantes para alcance dos objetivos da organização.
3. O descritivo é um texto resumido onde são caracterizados os objetos concretos e abstratos, seus atributos e relacionamentos.
4. Os objetos serão candidatos às entidades com seus atributos e relacionamentos.



## Os elementos da modelagem do banco de dados

---

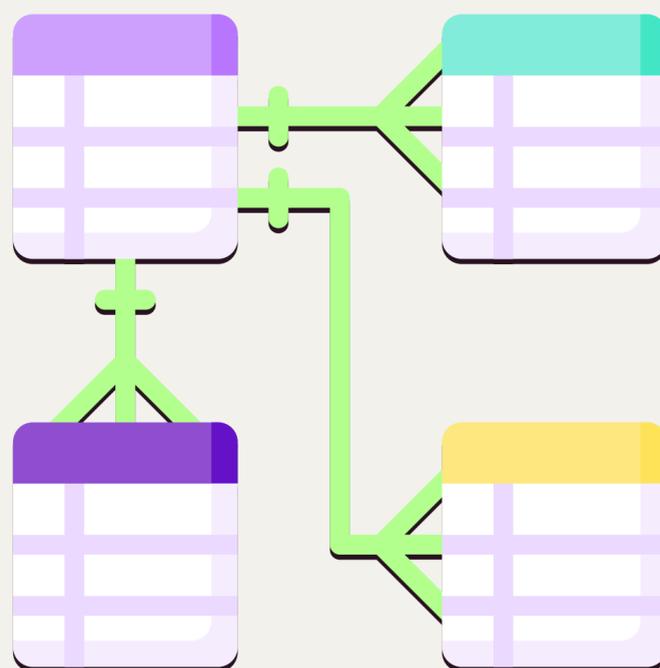
No modelo conceitual os elementos do BD importantes são as entidades, os atributos da entidades e os relacionamentos entre elas.

A leitura do descritivo textual após análise do domínio irá indicar os candidatos a elementos do modelo conceitual.

Os substantivos são candidatos a entidades. Estes têm atributos somente deles e não mais de outra entidade no mesmo domínio.

Os atributos são propriedades das entidades e podem ser descritivos (nome, endereço, telefone, descrição de produto, unidade física, quantidade em estoque, etc), ou identificadores (Cpf, matrícula, código, referência, número sequencial - desde que não sejam repetitivos).

Os relacionamentos entre entidades são associações e tem cardinalidades ou grau quantitativo de ocorrências de instância nas entidades. Exemplos: caneta e papel são ocorrências na entidade (tabela) de produtos. Maria e Marta são ocorrências na entidade passageiros.



## Entidades e atributos

---

São elementos importantes do modelo de dados. As entidades têm características próprias que são seus atributos. Importante saber:

Entidades representam os objetos do mundo real que serão as tabelas do banco de dados. Por exemplo, em um sistema de gerenciamento de biblioteca, "livro", "autor" e "leitor" são entidades. Cada entidade possui atributos, que são as características que descrevem a entidade. Um atributo pode ser um valor simples, como o título de um livro, ou um valor multivalorado como o endereço (rua, cidade, estado, país) de um autor.

**Entidade:** Um objeto do mundo real que será modelado no banco de dados.

**Atributo:** Uma propriedade ou característica do objeto que descreve como uma entidade.

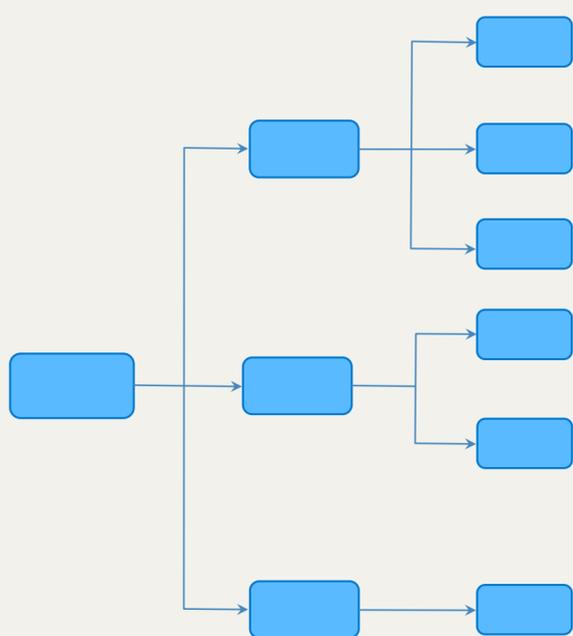
**Relacionamentos:** as entidades geralmente não existem de forma isolada. Elas interagem entre si através de associações. Por exemplo, um "livro" pode ser escrito por um "autor", e um "leitor" pode emprestar um "livro". O relacionamento entre entidades é fundamental para modelar a complexidade das informações.

Exemplo de relacionamento:

Uma pessoa tem um Cpf e um Cpf pertence a uma pessoa.

Numa compra a nota fiscal tem diversos produtos e um determinado produto está na nota fiscal.

Um livro pode ser escrito por vários autores. E um determinado autor pode escrever muitos livros.



## CAPÍTULO 2 – Modelagens conceitual e lógica

O processo de modelagem do banco de dados passa por duas fases complementares. A modelagem conceitual e a modelagem lógica.

A modelagem conceitual é uma abstração que descreve o domínio do mundo real que irá ser retratado através de informações estruturadas, prontas para acesso por sistemas programados.

A modelagem conceitual deve respeitar as regras do negócio e as restrições do processo prático. Exemplo da gestão de passagem aérea, que tem procedimentos, normas, restrições e modalidade de operações. Esses aspectos são definidos como requisitos funcionais da aplicação ou tarefas que o sistema irá executar, e que irá usar o banco de dados implementado após a modelagem lógica.

A modelagem lógica é esquematização do modelo conceitual onde são caracterizados os elementos técnicos para os relacionamentos e as cardinalidades entre as tabelas do modelo de dados, que por sua vez foram originadas das entidades e seus atributos do modelo conceitual.

A definição do modelo lógico está pela presença das chaves primárias referenciadas e das chaves estrangeiras que as referenciam, e os tipos de dados e tamanho de cada campo ou coluna das tabelas normalizadas em 3FN (terceira forma normal).

O Exemplo a seguir mostra a modelagem conceitual pela técnica da semântica e a realização da modelagem lógica.

# Descritivo de domínio para extrair elementos de BD

---

## Sistema de venda de passagens aéreas

No sistema de venda de **passagens** aéreas, **um cliente** compra **muitas** passagens, mas **só uma** passagem pertence a esse cliente. Passagem tem os atributos de **Número de cartão, tipo de cartão, poltrona, valor, número de voo**. Cliente tem **Cpf, nome, endereço e telefone**. A passagem pertence a **um voo** e **um** determinado voo pode **ter várias** passagens compradas. Voo **tem número, rota, dia e hora de saída, dia e hora de chegada**. Um voo é comandado por **um piloto** e **um** piloto pode **comandar muitos** voos num período. O piloto tem atributos de **matrícula, nome, telefone, endereço**. Um voo é **vinculado** a uma **aeronave**, que pode fazer **muitos** voos num período.

## Correspondência entre semântica e elemento do banco de dados

**Substantivos são candidatos a entidades**

**Adjetivos/substantivos são candidatos a atributos**

**Verbos/advérbios são candidatos a relacionamentos**

**Numeral são candidatos a cardinalidades**



# Elementos do Banco de Dados segundo a técnica da semântica

---

## Correspondência entre semântica e elemento do banco de dados

### Substantivos são candidatos a entidades

As entidades do modelo exemplo são **Cliente, Piloto, Passagem, Voo, Aeronave**. Foram detectadas por serem os objetos do domínio que originaram dos substantivos do descritivo.

### Adjetivos/substantivos são candidatos a atributos

Os atributos originados dos adjetivos/substantivos do descritivo do domínio, vinculados às entidades e atribuídos como suas propriedades ou características, são:

**Cliente(Cpf\_pk, Endereco, Telefone, Nome)**

**Piloto(Matricula\_pk, Nome\_piloto, Endereco\_piloto)**

**Passagem(Num\_cartao\_pk, Tipo\_cartao, Poltrona, Valor, Voo\_fk, Cpf\_fk)**

**Voo(Num\_voo\_pk, Data\_voo, Hora\_saida, Hora\_chegada, Rota, Matricula\_fk, Id\_fk)**

**Aeronave(Id\_pk, Modelo, Capacidade, Prefixo)**

### Verbos/advérbios são candidatos a relacionamentos

Os relacionamentos entre entidades originados dos verbos/advérbios, são: **Compra, Comanda, Pertence, Vinculado**.

### Numeral são candidatos a cardinalidades

As cardinalidades entre entidades originados das numerais, são: **Passagem->Cliente (1:n), Voo->Piloto (1:n), Voo->Aeronave (1:n), Passage->Voo(1:n)**.

# Sobre os atributos e chaves - elementos para o modelo lógico

---

## Atributos Simples e Compostos

Uma parte importante do processo de identificação dos atributos é distinguir entre atributos simples e atributos compostos. Atributos simples são aqueles que não podem ser subdivididos em partes menores. Por exemplo, o atributo nome de um produto é simples, pois é uma única propriedade que descreve o produto. Já os atributos compostos são aqueles que podem ser decompostos em vários subatributos. Um exemplo clássico de atributo composto seria o telefone de um cliente, que pode ser dividido em subatributos como ddi, ddd, número do telefone.

Essa diferenciação é importante porque, no modelo lógico, atributos compostos podem precisar ser transformados em múltiplas colunas ou até mesmo em tabelas auxiliares, caso sua decomposição seja necessária para melhor organização e consulta dos dados.

## Chave Primária

A chave primária é um campo ou conjunto de campos que identifica de forma única cada registro em uma tabela. Ela assegura que não existam registros duplicados ou ambíguos dentro da tabela, ou seja, cada linha deve ser única e facilmente identificável. Uma chave primária deve cumprir as seguintes condições:

**Unicidade:** O valor de uma chave primária deve ser único para cada registro da tabela. Isso impede que dois registros tenham o mesmo identificador.

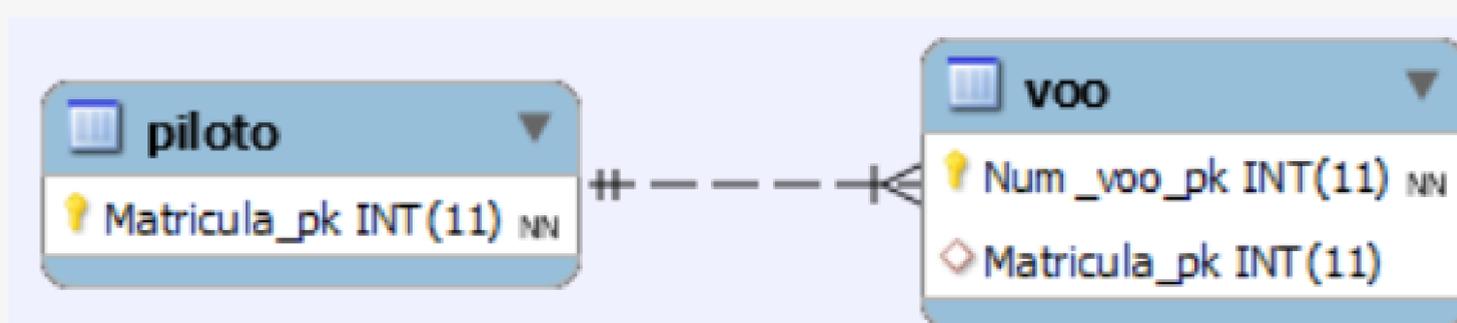
**Não Nulo:** Nenhum valor da chave primária pode ser nulo. Isso significa que todos os registros devem ter um valor válido para a chave primária.

**Índice Automático:** Em muitos SGBDs, a chave primária é automaticamente indexada ou auto-incrementada, o que acelera as buscas e outras operações de consulta.

## Exemplo de modelo lógico:

Na tabela produto, a chave primária é Id\_Produto. Esse campo identifica de forma única cada produto da tabela impedindo que dois produtos tenham o mesmo identificador. Isso é essencial para garantir a integridade dos dados, especialmente quando se estiver realizando operações de busca ou atualizações.

Na tabela piloto, a chave primária é matricula\_id, que garante que cada piloto inserido tenha um identificador único.



## Chave Estrangeira

A chave estrangeira é um campo ou conjunto de campos em uma tabela que faz referência à chave primária de outra tabela. Seu principal objetivo é garantir a integridade referencial entre as tabelas, ou seja, assegurar que os dados em uma tabela sejam consistentes com os dados em outra. A chave estrangeira estabelece relacionamentos entre as tabelas e é fundamental para a normalização do banco de dados.

A chave estrangeira pode ter as seguintes características:

- **Integridade Referencial:** A chave estrangeira deve ter um valor que corresponda a uma chave primária existente na tabela referenciada. Caso contrário, o banco de dados impede a inserção ou atualização desse valor, mantendo a consistência entre as tabelas.
- **Relacionamento entre Tabelas:** As chaves estrangeiras criam vínculos entre as tabelas, permitindo operações como JOIN e garantindo que as tabelas permaneçam interconectadas de maneira lógica e eficiente.

## CAPITULO 3 – Diagramas conceitual e lógico

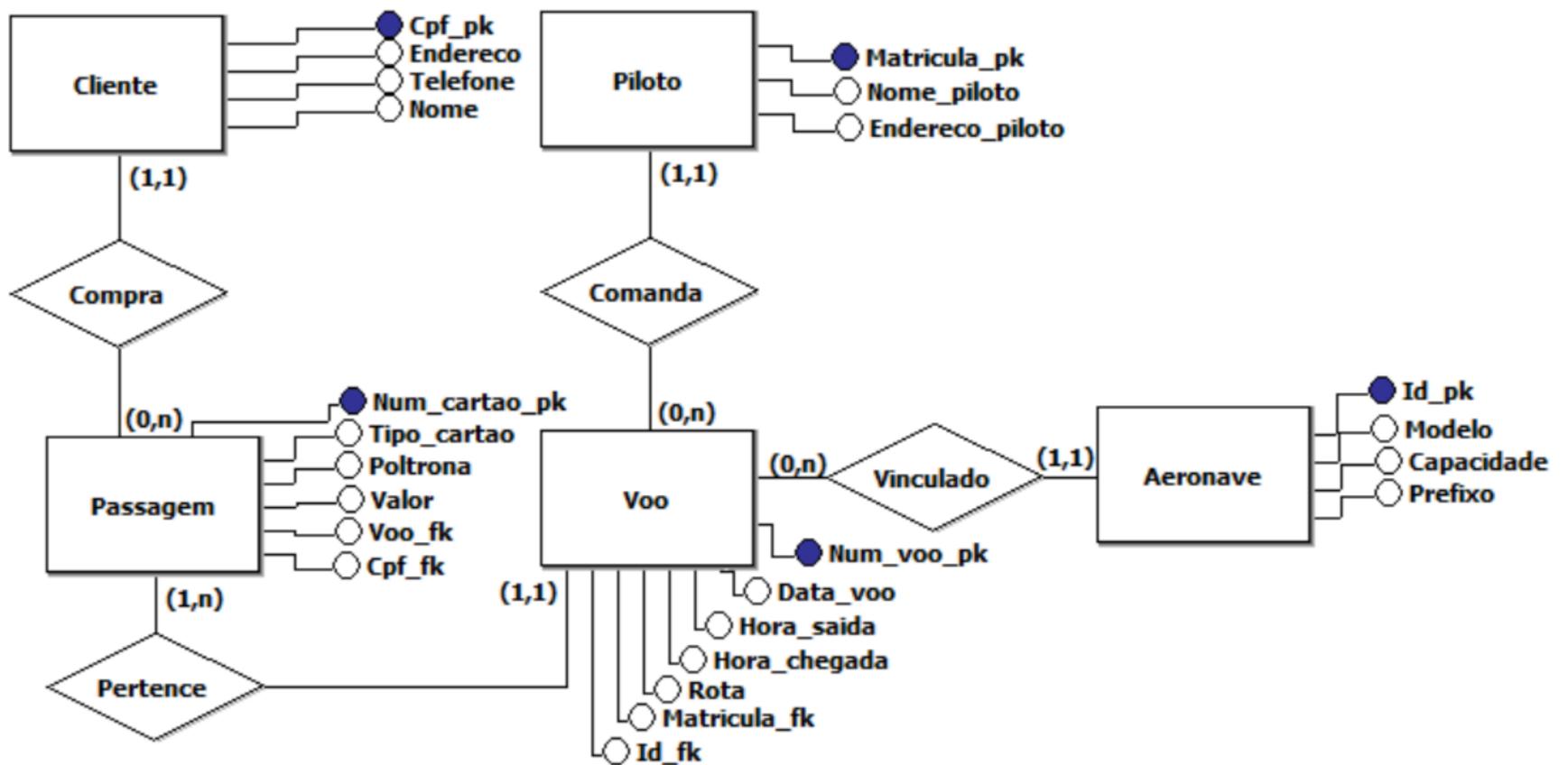
Diagrama é um gráfico de esquemas com elementos representativos do domínio e é mostrado resumidamente ou simplificado. Utiliza símbolos geométricos como retângulo, triângulo, losango e linha que correspondem aos elementos do banco de dados, como entidades, relacionamentos e chaves.

A modelagem conceitual é representada pelo Diagrama Entidade-Relacionamento (DER) no modelo conceitual. É um gráfico que deve espelhar o domínio do banco de dados na mais alta abstração e com seus elementos entidades e relacionamentos mostrados conceitualmente sem a preocupação de esquemas técnicos ou lógicos. Nele não precisa necessariamente conter chaves identificadoras, tipos de dados ou tamanho dos atributos.

O DER também representa um esquema lógico, usa-se os mesmos elementos do modelo conceitual, mais as cardinalidades (mínima, máxima), as chaves primárias e estrangeiras, tipo e tamanho dos dados dos atributos. Deve-se estar na 3ª Forma Normal (3FN) e sem apresentar relacionamentos em generalização/especialização, nem cardinalidade muitos para muitos (n:m), pois esta já foi convertida em (1:n) com a inclusão da terceira tabela associativa.

# Modelo Conceitual - Diagrama Entidade-Relacionamento

Sistema de venda de passagens aéreas



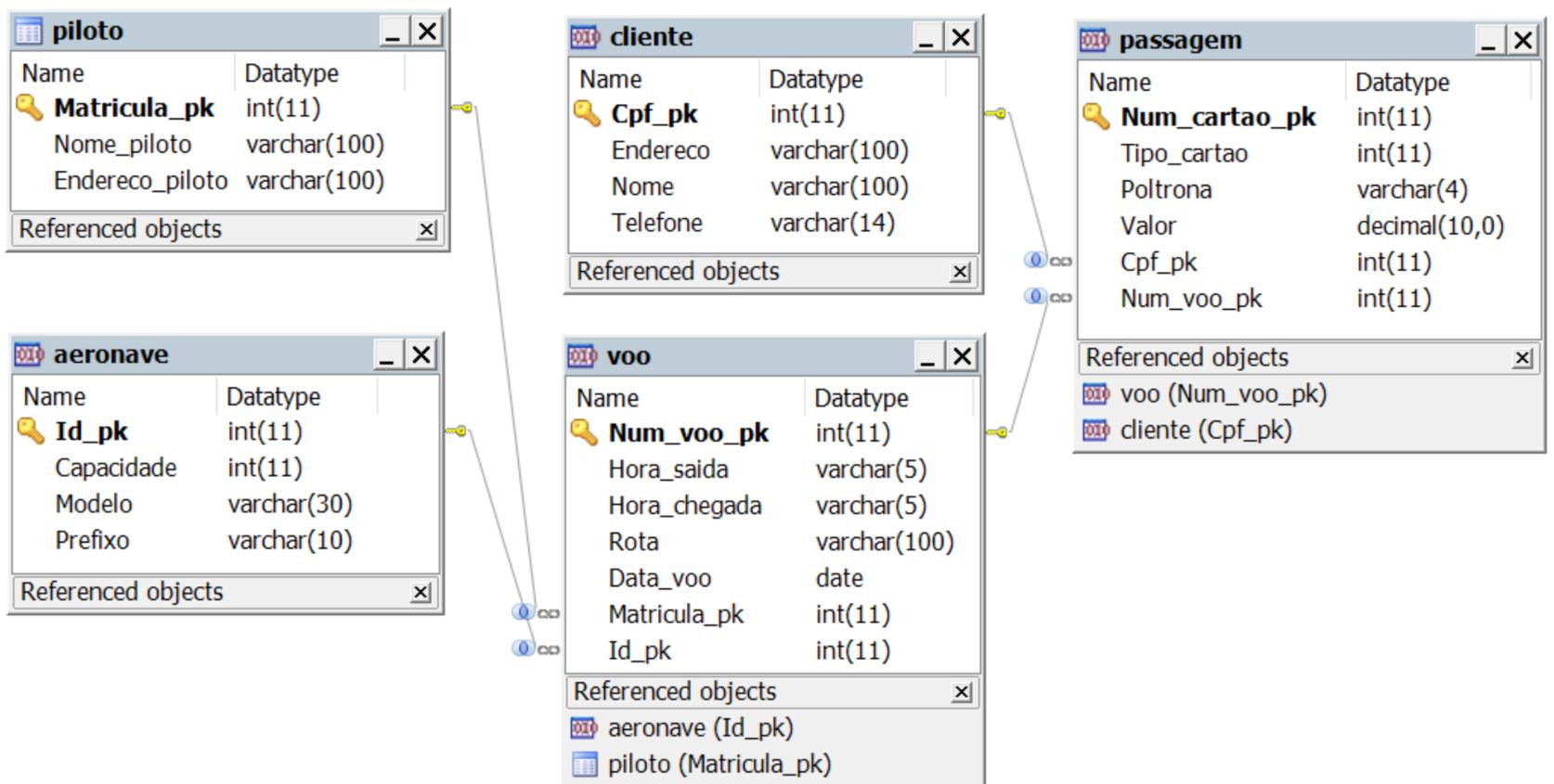
## Métodos mapeamento modelo conceitual para o lógico:

A passagem do modelo conceitual para o lógico segue os seguintes passos:

1. Confirmar entidades e seus atributos do modelo conceitual para 3FN.
2. Definir as chaves primárias e estrangeiras para cada entidade.
3. Identificar relacionamentos e mapear as cardinalidades mínima-máxima.
4. Definir tipos e tamanhos de dados para os atributos.
5. Resolver relacionamentos (1:1, 1:n, n:m). Quando (muitos - para-muitos, n:m) criar tabela intermediária que no modelo lógico será (1:n).

## DER do modelo lógico do BD exemplo

### Sistema de venda de passagens aéreas



A Figura acima mostra um DER lógico com entidades e atributos do modelo do banco de dados normalizado na 3FN.

O que caracteriza o modelo lógico e diferencia do modelo conceitual é a presença das chaves primária e estrangeiras, referenciadas segundo o relacionamento e as cardinalidades (1:1, 1:n ou n:m) e a caracterização dos atributos com tipo de dado e o tamanho de dados.

### Exemplo Rota: Varchar(250) ou Data\_voo: Date

Cada entidade possui atributos qualitativos que são as características ou propriedades que descrevem essas entidades e serão as colunas das tabelas no modelo físico.

Por exemplo, uma entidade aeronave pode ter atributos como id\_pk: int, capacidade: int e modelo: varchar(30). Cada um desses atributos é uma característica única que identifica uma aeronave de maneira distinta dentro do sistema, e definem o tipo de dados e o tamanho do dados que irão conter.

## A implementação física do banco de dados:

O banco de dados **passagem\_aerea** foi implementado usando o SGBD MySQL 5.0.

A SQL aplicada no código abaixo é uma linguagem declarativa projetada para realizar operações como criação, consulta, atualização e exclusão de dados, além de gerenciar estruturas e permissões dentro do banco de dados. É utilizada em sistemas como MySQL, PostgreSQL, Oracle Database e Microsoft SQL Server. A SQL se consolidou como um padrão internacional reconhecido pela ANSI (American National Standards Institute) e pela ISO (International Organization for Standardization).

```
CREATE database passagem_aerea;
```

```
CREATE TABLE Aeronave (  
  Id_pk Int PRIMARY KEY,  
  Capacidade Int,  
  Modelo Varchar(30),  
  Prefixo Varchar(10)  
);
```

```
CREATE TABLE Cliente (  
  Cpf_pk Int PRIMARY KEY,  
  Endereco Varchar(100),  
  Nome Varchar(100),  
  Telefone Varchar(14)  
);
```

```
CREATE TABLE Piloto (  
  Matricula_pk Int PRIMARY KEY,  
  Nome_piloto Varchar(100),  
  Endereco_piloto Varchar(100)  
);
```

```
CREATE TABLE Passagem (  
  Num_cartao_pk Int PRIMARY KEY,  
  Tipo_cartao Int,  
  Poltrona Varchar(4),  
  Valor Decimal,  
  Cpf_pk Int,  
  Num_voo_pk Int,  
  FOREIGN KEY(Num_voo_pk) REFERENCES  
  Voo (Num_voo_pk),  
  FOREIGN KEY(Cpf_pk) REFERENCES Cliente  
  (Cpf_pk)  
);
```

```
CREATE TABLE Voo (  
  Num_voo_pk Int PRIMARY KEY,  
  Hora_saida Varchar(5),  
  Hora_chegada Varchar(5),  
  Rota Varchar(100),  
  Data_voo Date,  
  Matricula_pk Int,  
  Id_pk Int,  
  FOREIGN KEY(Id_pk) REFERENCES Aeronave (Id_pk),  
  FOREIGN KEY(Matricula_pk) REFERENCES Piloto (Matricula_pk)  
);
```

# CAPÍTULO 4 - Normalização

## **Sobre a normalização, integridade referencial e BD consistente.**

**Normalização:** é o processo para se reconhecer tabelas embutidas em tabelas, ou seja, quando há tabelas internas em tabelas externas. Com atributos em falta com dependência funcional - atributo que não é representado pela chave primária ou quando o atributo apresenta transitividade direta entre atributos não chaves - atributo descritivo que não é chave mas representa outro atributo descritivo na mesma tabela.

Exemplo: **Autor(Cpf, Nome, Telefone, Isbn, TituloLivro, EdicaoLivro)**

Ao dividir os dados em tabelas distintas e usar chaves estrangeiras para relacioná-las, é possível evitar a redundância de dados e manter o banco de dados consistente e com integridade referencial entre tabelas.

O processo de normalização apresenta-se em fases conhecidas com Formas Normais, que recebe a notação 1FN (primeira forma normal), 2FN, 3FN, ... 7FN. Um banco de dados ou uma tabela quando está na 3FN considera-se normalizado, acima dessa fase as formas normais prescritas são para revisões ou manutenção da modelagem.

Formas normais são hierárquicas ou seja, só irá para 1FN se a tabela está NN (não normalizada). Só irá para a 2FN se a 1FN estiver realizada, assim por diante.

# Sobre os atributos e chaves - elementos para o modelo lógico

---

**Restrição de Atualização e Exclusão:** É possível definir comportamentos específicos ao modificar ou excluir registros. Por exemplo, ao deletar um registro em uma tabela referenciada, pode-se definir que todos os registros relacionados sejam também deletados (on delete cascade), ou que a chave estrangeira seja definida como nula (on delete set null).

**Integridade referencial com chaves primárias e estrangeiras no modelo lógico de dados:**

**Integridade dos Dados:** As chaves primárias garantem que os registros dentro de uma tabela sejam únicos, enquanto as chaves estrangeiras garantem que as relações entre as tabelas sejam válidas e consistentes.

**Facilidade nas Consultas:** O uso de chaves primárias e estrangeiras facilita operações de consulta, permitindo realizar junções eficientes entre tabelas relacionadas.

**Manutenção de Dados:** As chaves estrangeiras quando configuradas corretamente com regras de atualização e exclusão (ex.: on delete cascade), asseguram que as alterações em uma tabela sejam refletidas nas tabelas relacionadas de maneira consistente.

# Relacionamentos e cardinalidades entre tabelas

---

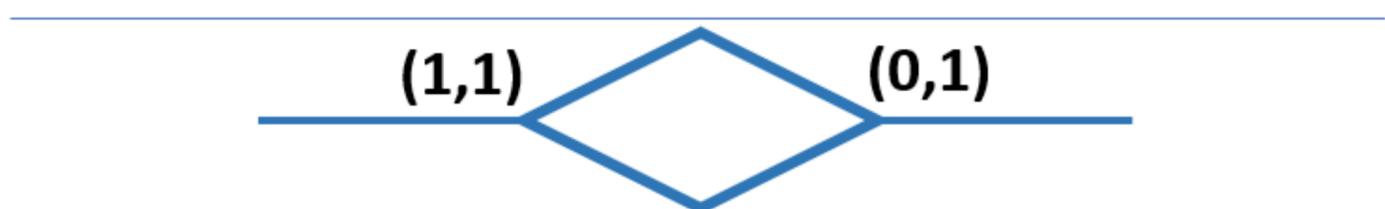
## Cardinalidades nos relacionamentos entre tabelas

As cardinalidades definem as relações numéricas entre as entidades, ou seja, quantas ocorrências de uma tabela podem ou devem se associar a ocorrências de outra tabela. A correta definição das cardinalidades é fundamental para garantir que o relacionamento entre tabelas seja eficaz, evitando redundâncias e inconsistências. Utilizam as notações máxima (1:n) e mínima-máxima (1,1):(1:n).

As cardinalidades mais comuns são:

### 1:1 (Um para Um)

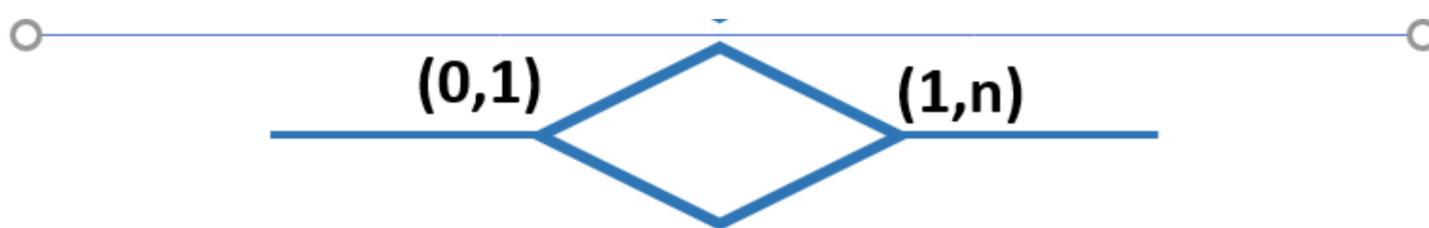
Na cardinalidade 1:1, cada registro de uma entidade está associado a exatamente um registro da outra entidade. Embora menos comum, essa relação pode ser usada em situações específicas. No modelo lógico, um exemplo de relacionamento 1:1 poderia ser entre Passageiro e Poltrona do avião. Cada passageiro é alocado a uma única poltrona, e cada poltrona possui um único passageiro alocado. Esse tipo de relacionamento poderia ser representado no modelo lógico pela inclusão de uma chave primária em ambas as tabelas passageiro e poltrona, onde uma delas poderia atuar como chave estrangeira na outra, garantindo que ambas as entidades compartilhem a mesma instância de dados.



## 1:n (Um para Muitos)

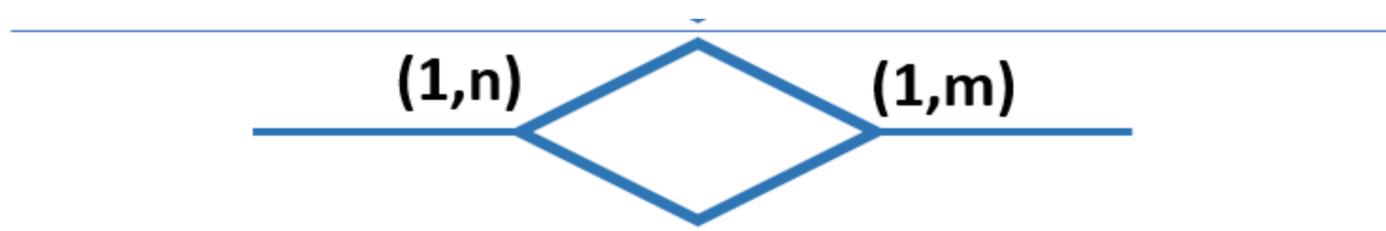
A cardinalidade 1:n indica que um único registro de uma entidade pode se relacionar com vários registros de outra entidade. Esse tipo de relacionamento é muito comum em bancos de dados relacionais. No modelo de passagem aérea que está neste livro, um cliente pode adquirir várias passagens, mas cada passagem é comprada por um único passageiro.

Esse tipo de relacionamento é frequentemente representado por uma chave estrangeira na tabela "muitos", apontando para a chave primária da tabela "um". Por exemplo, a tabela passagem teria uma chave estrangeira que faz referência à chave primária de cliente, representando o vínculo entre essas duas entidades.



## n:m (Muitos para Muitos)

A cardinalidade n:m é mais complexa, pois envolve múltiplas ocorrências de uma entidade se relacionando com múltiplas ocorrências de outra entidade. No modelo considerado, um exemplo de n:m ocorre entre piloto e aeronave, onde um piloto pode operar várias aeronaves num período, e uma aeronave pode ser operada por vários pilotos. Esse tipo de relacionamento não pode ser diretamente implementado em uma tabela relacional sem uma terceira tabela de junção. Nesse caso, a tabela de junção é voo, que é um subdomínio existente no domínio passagem aérea. Mas poderia ser uma tabela que não representasse um subdomínio real e servisse apenas para associação, então seria criada uma terceira tabela piloto\_aeronave para representar essa cardinalidade.



Para resolver esse relacionamento no modelo lógico, a tabela de junção voo armazena as chaves primárias das duas entidades e pode incluir atributos adicionais, como data de emissão da passagem ou outros dados pertinentes ao relacionamento.

## Tipos e Tamanhos de Dados

A escolha adequada dos tipos e tamanhos de dados em um banco de dados é essencial para garantir eficiência no armazenamento, performance e integridade dos dados. Tipos numéricos como TINYINT, INT e BIGINT são usados para armazenar valores inteiros, enquanto DECIMAL, FLOAT e DOUBLE são usados para números com casas decimais, dependendo da precisão necessária. Para armazenar textos, utiliza-se CHAR para tamanho fixo e VARCHAR para textos de comprimento variável. TEXT é utilizado para grandes blocos de texto.

Tipos temporais como DATE, TIME, DATETIME e TIMESTAMP armazenam informações de data e hora, sendo úteis para registros e históricos. BOOLEAN é usado para armazenar valores lógicos como verdadeiro ou falso. Os tipos binários como BINARY e BLOB são usados para armazenar dados não textuais, como imagens e vídeos. Além disso, existem tipos especiais como ENUM, para valores predefinidos, e JSON ou XML, para dados estruturados ou semi-estruturados, como documentos ou configurações. A escolha correta desses tipos contribui para uma modelagem eficiente e para o bom desempenho do banco de dados.

Dados Numéricos Inteiros		
Tipo	Escopo com sinal	Escopo sem sinal
Tinyint	-128 a 127	0 a 255
Smallint	-32.768 a 32.767	0 a 65.535
Mediumint	-8.388.608 a 8.388.607	0 a 16.777.215
Int	-2.147.483.648 a 2.147.483.647	0 a 4.294.967.295
Bigint	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807	0 a 18.446.744.073.709.551.615

Para atributos que armazenam nomes, como o nome de um produto ou cliente, o tipo de dado VARCHAR é uma boa escolha, pois ele permite armazenar textos de tamanho variável. Por exemplo, se um campo de Nome for definido como VARCHAR(255), ele pode conter qualquer texto até 255 caracteres, sem desperdiçar espaço se o nome for menor.

Quando precisamos armazenar valores numéricos, como o valor de um passagem o tipo DECIMAL é ideal. Usando o formato DECIMAL(10,2), por exemplo, podemos armazenar um número com até 10 dígitos no total, sendo 2 desses dígitos após a vírgula. Isso é ideal para valores monetários, garantindo que os números sejam armazenados com precisão sem arredondamentos errados.

Para informações como telefone, que podem incluir números e caracteres especiais, como parênteses e hifens, o tipo VARCHAR(14) é uma escolha adequada. Ele armazena até 14 caracteres, o que é suficiente para números de telefone em diversos formatos.

Se precisar registrar datas, como a data de uma solicitação de serviço, o tipo DATE deve ser utilizado. Ele armazena apenas a data (ano, mês e dia), sem a necessidade de incluir a hora. Esse tipo é muito útil para situações em que a hora não é necessária, como ao registrar quando um evento ocorreu.

<b>Dados de Texto Não-Binário</b>	
<b>Tipo de texto</b>	<b>Numero máximo de bytes</b>
Tinytext	255
Text	65.535
MediumText	16.777.215
LongText	4.294.967.295
Varchar	65.535
Char	255

# Referências

DATE, C. J. Introdução a Sistemas de Banco de Dados. Tradução da 7ª. Edição Americana. Editora Campus.

DATE, C.J. Bancos de Dados, Tópicos Avançados, Editora Campus

KORTH, H. F., SILBERSCHATZ, A. Sistema de Banco de Dados, Makron Books

GARCIA-MOLINA, H. ULLMAN, J.D., WIDOM, J. Implementação de Sistemas de Bancos de Dados. Editora Campus

MACHADO, Felipe Nery Rodrigues. Banco de Dados – projeto e implementação. Érica, 2004. São Paulo

