



LICENCIATURA

EM

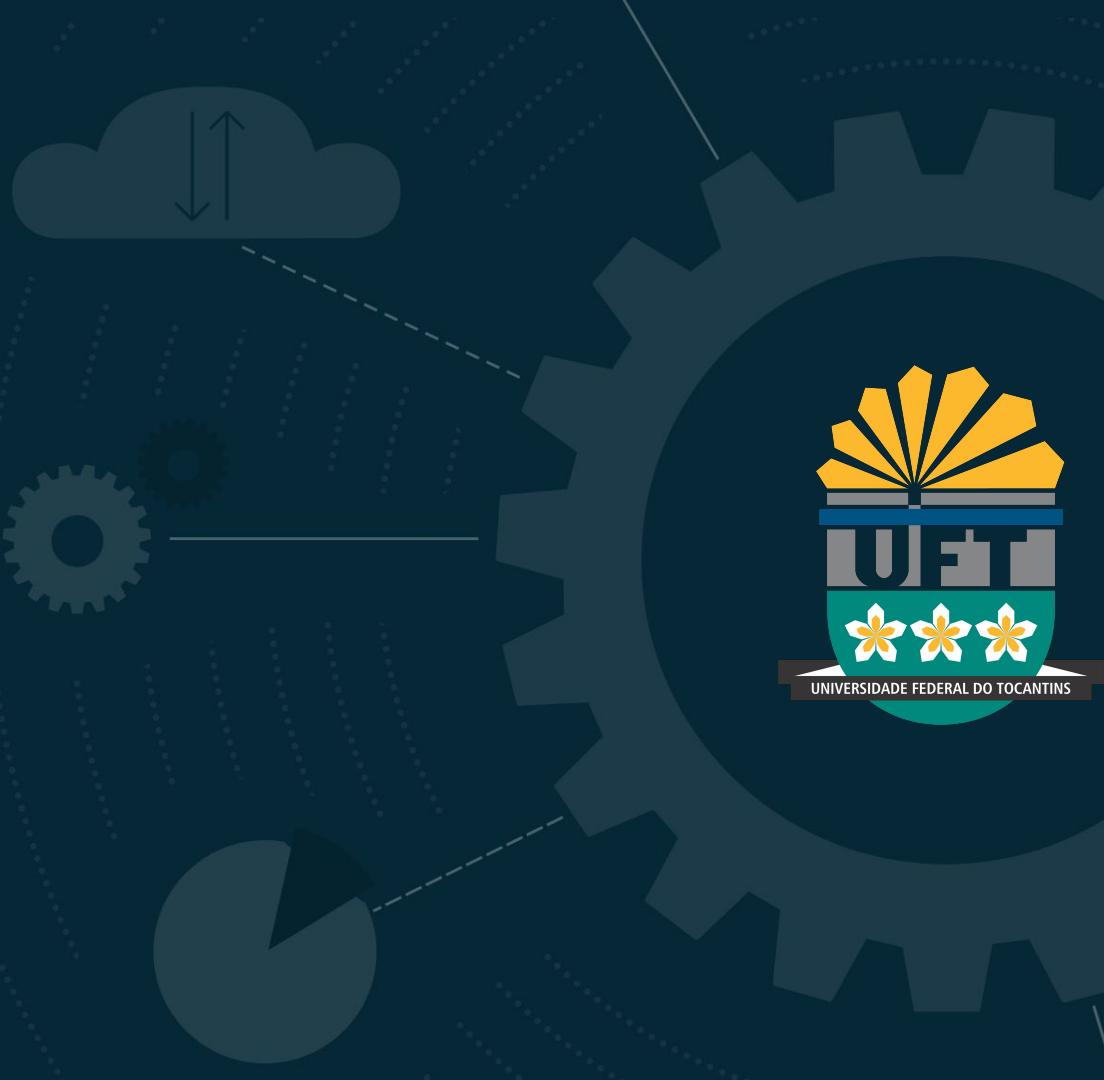
# COMPUTAÇÃO

# Banco de Dados

Prof. Dr. Eduardo Ribeiro

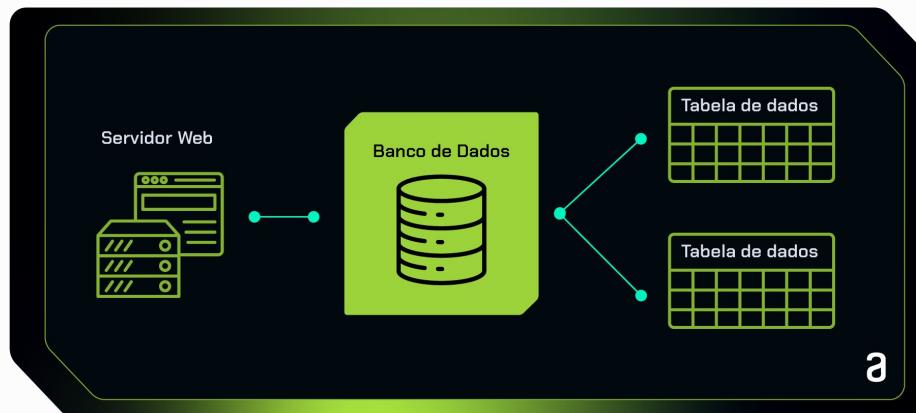


# Introdução aos Bancos de dados



# O que veremos nessa aula?

- Diferença entre Dado e Informação
- Tipos de Dados
- Bancos de Dados e Seus Conceitos
- Propriedades ACID
- Histórico dos Modelos de Dados
- Introdução ao Modelo Relacional



# Introdução

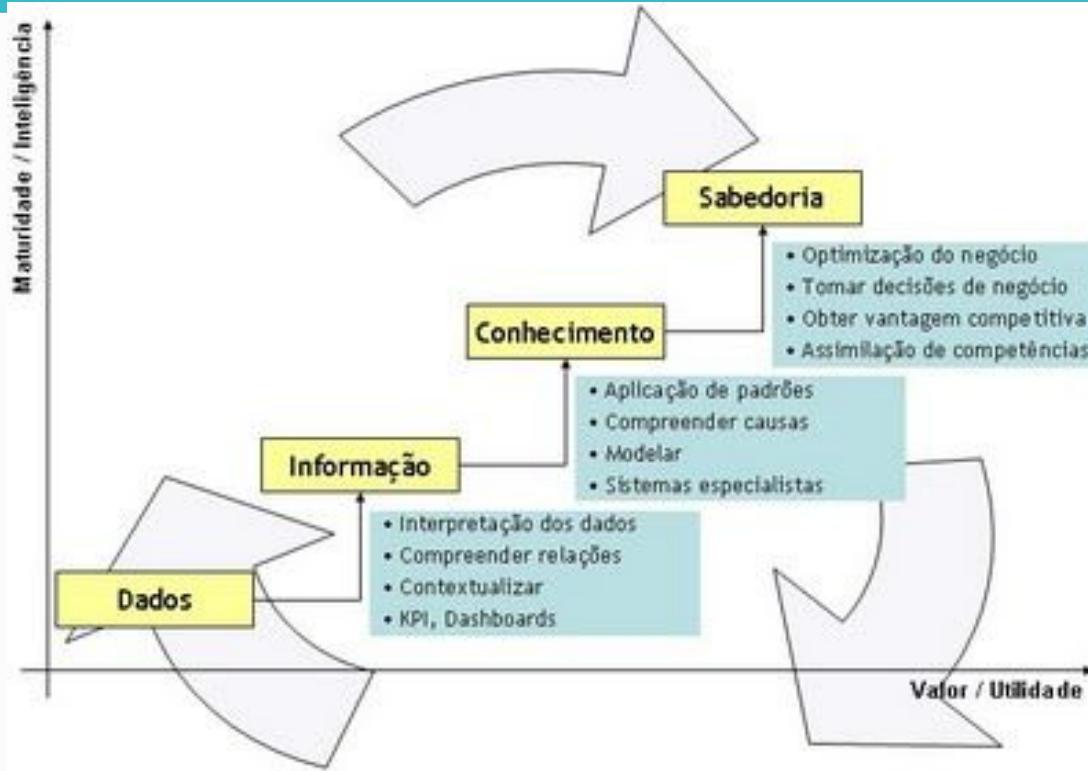
- **Introdução aos Bancos de Dados**
  - Organização e significado dos dados
  - Transformação de dados brutos em informações úteis
- **Fundamentos Essenciais**
  - Compreensão da estrutura de dados digitais
  - Distinção entre Dados e Informação
- **Importância da Informação**
  - Capacita a tomada de decisões informadas
  - Implementação de estratégias eficazes
- **Objetivo da aula**
  - Estabelecer uma base sólida para o estudo de bancos de dados
  - Ressaltar a importância de transformar dados em ativos estratégicos para organizações modernas



# Diferença entre Dado e Informação

- **Diferença entre Dados e Informação**
  - Dados: valores brutos e primários, sem significado específico ou contexto.
  - Informação: dados organizados, processados e interpretados para fornecer valor e conhecimento.
- **Exemplos Práticos de Transformação de Dados em Informação**
  - Gestão de Clientes: dados brutos convertidos em informações para segmentação e comunicação.
  - Controle de Estoque: dados organizados para monitorar e prever necessidades.
  - Registro de Funcionários: dados estruturados para cálculos e avaliações.
- **Importância da Transformação de Dados em Bancos de Dados**
  - Facilita a extração de valor significativo dos dados armazenados.
  - Bancos de dados são projetados para organizar e recuperar dados eficientemente, gerando informações úteis.





# Tipos de Dados

- **Categorias de Dados e sua Importância**

- Dados podem ser estruturados, semiestruturados ou não estruturados.
- Escolher a categoria correta é essencial para o armazenamento e manipulação eficaz em bancos de dados.

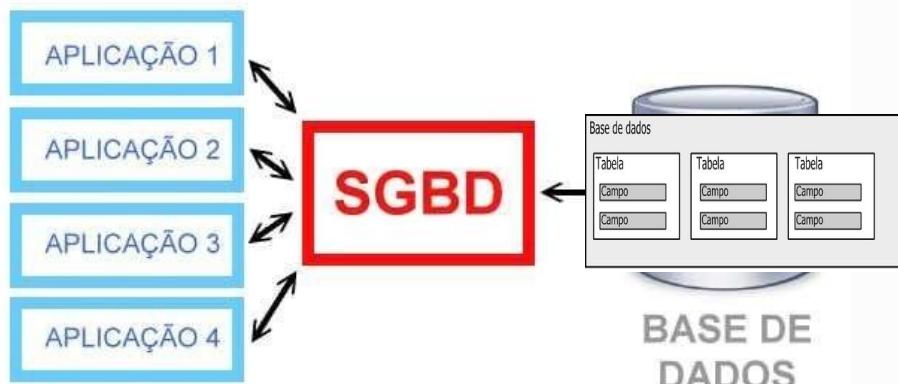


# Tipos de Dados

- **Dados Estruturados:**

- Organização: Armazenados em formatos fixos e predefinidos, geralmente em tabelas com linhas e colunas.
- Acesso: Facilitam consultas e análises através de linguagens estruturadas como SQL.
- Exemplos:
  - Informações em bancos de dados relacionais, como registros de clientes ou transações financeiras.

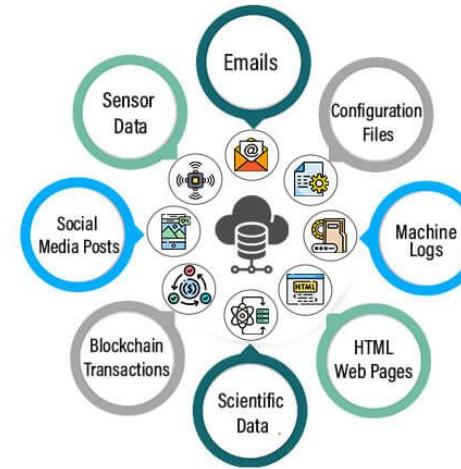
Nome	CPF	Endereço	Telefone
Marcela Freitas	11111	Rua A, nº 1	101010
João Augusto	22222	Rua B, nº 2	202020
Pablo Silva	33333	Rua C, nº 3	303030
André Mendes	44444	Rua D, nº 4	404040
Juliana Freitas	55555	Rua E, nº 5	505050



# Tipos de Dados

- **Dados Semiestruturados:**

- Organização: Não possuem uma estrutura rígida como os dados estruturados, mas têm algum nível de organização, como tags ou marcadores que proporcionam algum contexto.
- Acesso: Permitem consultas flexíveis, embora não tão eficientes quanto dados estruturados, usando linguagens como XPath ou JSONPath.
- Exemplos:
  - Documentos XML ou JSON, onde cada documento pode ter estrutura variável, mas com elementos definidos.



```
<note>
  <date>
    <year>2008</year>
    <month>01</month>
    <day>10</day>
  </date>
  <to>Tove</to>
  <from>Jani</from>
</note>
```

# Tipos de Dados

- **Dados Não Estruturados:**

- Organização: Não possuem estrutura organizada definida, sendo frequentemente descritos como dados "não formatados".
- Acesso: Difíceis de consultar e analisar diretamente devido à falta de estrutura, exigindo técnicas avançadas como processamento de linguagem natural ou reconhecimento de padrões.
- Exemplos:
  - Arquivos de mídia, como imagens, vídeos, e-mails, onde não há formato padronizado para armazenamento de dados.



# Tipos de Dados

## Diferença entre Tipos de Dados

### Dados Estruturados

Pode ser exibido em linhas, colunas e banco de dados relacional



Números, datas e caracteres



Estimativa de 20% dos dados corporativos (Gartner)



Requer pouco armazenamento



Mais fácil de gerenciar e proteger com soluções legadas



### Dados Não Estruturados

Não pode ser exibido em linhas, colunas e banco de dados relacional



Imagens, áudios, vídeos, arquivos de processamento de textos e planilhas eletrônicas



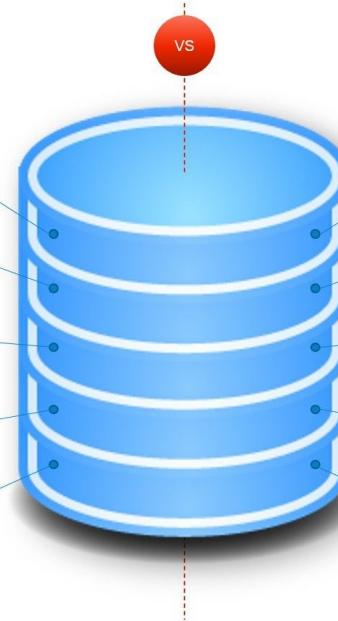
Estimativa de 80% dos dados corporativos (Gartner)



Requer grande armazenamento

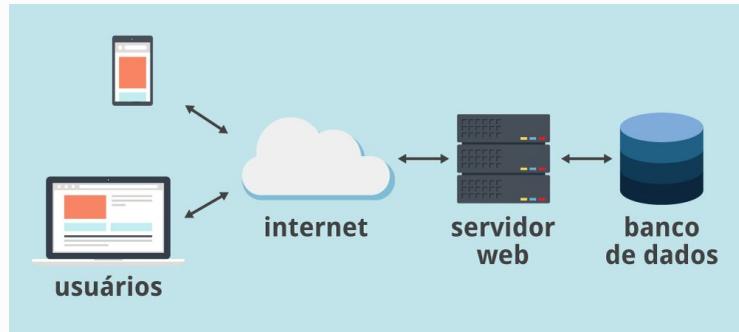


Mais difícil de gerenciar e proteger com soluções legadas



# Bancos de Dados Estruturados e Seus Conceitos

- Bancos de dados organizam grandes volumes de dados de forma lógica e coerente.
- Facilitam a transformação de dados em informações úteis para a tomada de decisões.
- Conceitos Fundamentais de Bancos de Dados
  - Tabela: Estrutura que armazena dados em linhas (registros) e colunas (campos).
  - Registro: Linha em uma tabela que contém dados relacionados.
  - Campo: Coluna em uma tabela que representa um atributo específico dos dados.
- Chaves em Bancos de Dados
  - Chave Primária: Identifica de forma única cada registro em uma tabela.
  - Chave Estrangeira: Campo que referencia a chave primária de outra tabela, estabelecendo relações e integridade referencial.



O diagrama mostra uma tabela com 6 linhas e 3 colunas. As colunas são rotuladas como 'Coluna (atributo)' e as linhas como 'Linha (tupla)'. A tabela é intitulada 'Pacientes' e contém os seguintes dados:

Paciente	CPF	Tipo Sanguíneo
Marcela Freitas	11111	A+
João Augusto	22222	B-
Pablo Silva	33333	AB+
André Mendes	44444	O-
Juliana Freitas	55555	A-

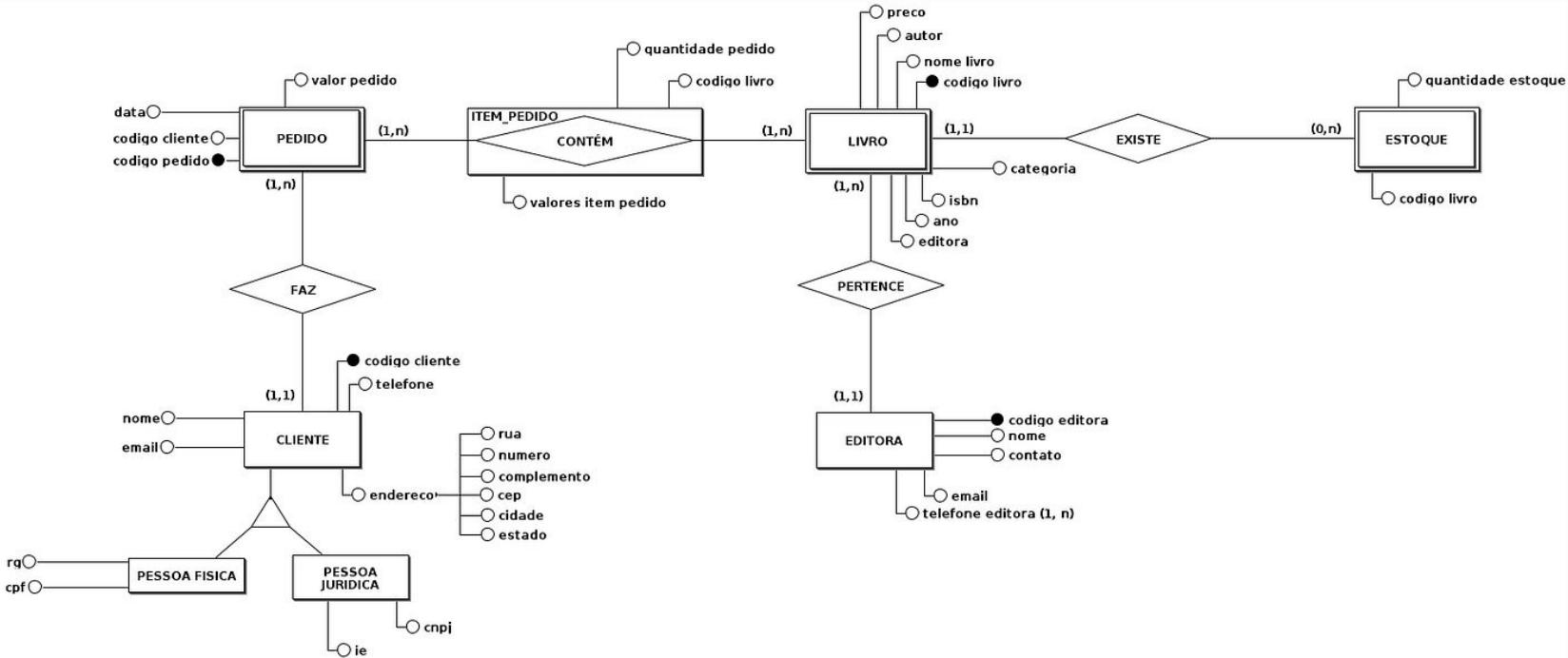
Um círculo vermelho redondo circunda a linha de João Augusto. Dentro desse círculo, uma seta aponta para o valor '22222' na coluna 'CPF', com a legenda 'valor do atributo'. Outra seta aponta para a coluna 'CPF' com a legenda 'Coluna (atributo)'. Um círculo vermelho redondo circunda a coluna 'CPF', com a legenda 'Linha (tupla)'.



# Exemplos de Bancos de Dados

- **BD de Gestão de Recursos Humanos:**
  - Finalidade: Armazenamento de informações sobre funcionários, como nome, endereço, cargo, salário e histórico de desempenho.
  - Exemplo: Uma tabela de "Funcionários" em um Tribunal de Contas pode incluir campos como "Nome", "Lotação", "Carga Horária", "CPF", etc.
- **BD de Sistemas de Vendas e Comércio:**
  - Rastreamento de produtos, vendas, inventários e clientes.
  - Exemplo: Uma tabela de "Vendas" em uma empresa de comércio eletrônico pode incluir campos como "ID do Produto", "Nome do Produto", "Quantidade Vendida", "Data da Venda", "Preço Unitário".
- **BD de Sistema de Biblioteca:**
  - Finalidade:
  - Exemplo:
- **BD de Livraria:**
  - Finalidade:
  - Exemplo:

# EXEMPLO: Banco de Dados de uma Livraria



# Diferença entre Planilhas do Excel e Bancos de Dados

- **Planilhas do Excel:**

- Adequadas para um único usuário ou poucos usuários.
- Não facilitam modificações complexas nos dados.
- Armazenam dados relativamente menores e mais simples.

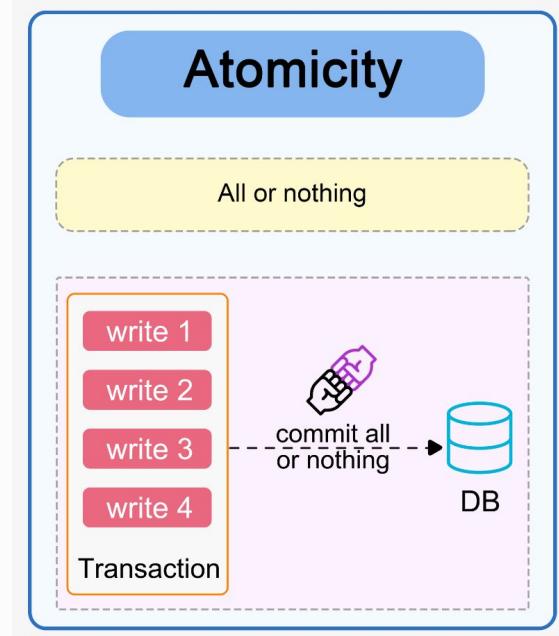
- **Bancos de Dados:**

- Projetados para uso por múltiplos usuários simultaneamente.
- Utilizam linguagens e lógicas altamente complexas para modificar, atualizar e consultar os dados.
- Armazem enormes quantidades de informações estruturadas e complexas.



# Propriedades ACID

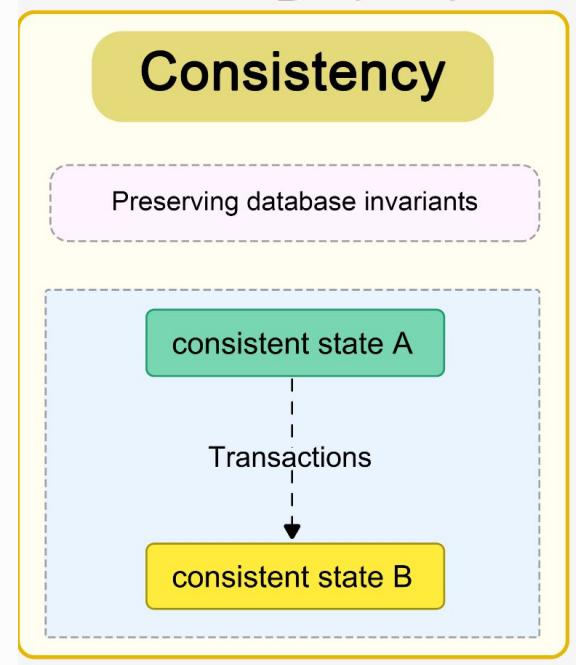
- ACID é um conjunto de princípios essenciais para garantir a integridade e confiabilidade das operações em bancos de dados.
- Composto por Atomicidade, Consistência, Isolamento e Durabilidade, ACID é fundamental para o gerenciamento de transações.
- **Atomicidade (Atomicity)**
  - Garante que uma transação seja tratada como uma unidade indivisível.
  - Todas as operações da transação devem ser concluídas com sucesso para que seja considerada bem-sucedida.
  - Exemplo prático:
    - Imagine que você está realizando uma transferência bancária entre duas contas. A transação envolve duas etapas:
      - Retirar dinheiro da conta A.
      - Depositar dinheiro na conta B.
    - A atomicidade assegura que ambas as etapas sejam concluídas com sucesso ou nenhuma delas será realizada. Portanto, não haverá um momento em que o dinheiro seja retirado da conta A sem ser depositado na conta B, evitando inconsistências e perdas de dinheiro.



# Propriedades ACID

- **Consistência (Consistency)**

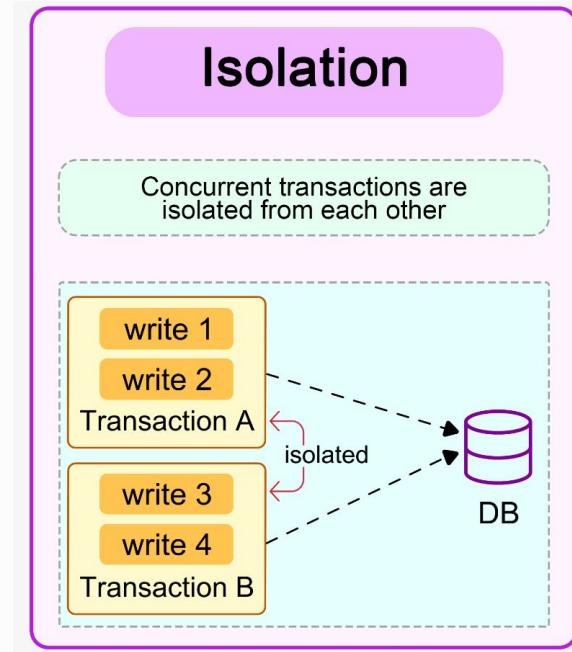
- A consistência assegura que uma transação leve o banco de dados de um estado válido para outro estado válido.
  - Isso significa que qualquer transação realizada deve respeitar todas as regras e restrições do banco de dados, garantindo que os dados permaneçam corretos e integrados após a conclusão da transação.
- Exemplo Prático:
  - Usando o exemplo da transferência bancária, a soma dos saldos das contas A e B deve ser a mesma antes e depois da transação.
  - Se antes da transferência a soma dos saldos era \$1000, após a transferência, a soma deve continuar sendo \$1000.
  - Se a transação não puder manter essa consistência, ela será revertida ao estado anterior.



# Propriedades ACID

- **Isolamento (*Isolation*)**

- O isolamento garante que as transações concorrentes sejam executadas de forma que não interfiram umas nas outras.
- Cada transação deve operar como se fosse a única em execução no sistema, evitando que as operações intermediárias de uma transação sejam visíveis para outras transações.
- Exemplo Prático:
  - Suponha que João e Maria compartilham uma conta bancária e ambos tentam sacar dinheiro ao mesmo tempo.
  - O isolamento garante que as transações de saque sejam processadas uma de cada vez.
  - Se João inicia um saque de \$50 enquanto Maria saca \$30 simultaneamente, o banco de dados processará uma transação completamente antes de iniciar a outra.
  - Isso evita que uma transação veja dados incompletos ou inconsistentes de outra transação em andamento.



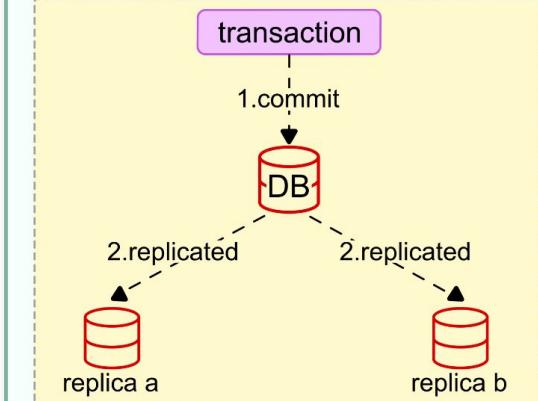
# Propriedades ACID

- **Durabilidade (Durability)**

- A durabilidade assegura que, uma vez que uma transação foi confirmada como concluída, suas alterações são permanentes e não serão perdidas, mesmo em caso de falhas no sistema, como quedas de energia ou bugs.
- A durabilidade é geralmente implementada através do uso de logs de transações e backups.
- Exemplo:
  - Voltando à transferência bancária, após a conclusão bem-sucedida da transação, o novo saldo nas contas A e B é gravado no banco de dados.
  - Se ocorrer uma falha no sistema logo após a transação, os novos saldos devem ser preservados.
  - Técnicas como logs de transações (que registram todas as operações realizadas durante uma transação) e backups garantem que os dados sejam recuperáveis e consistentes após a recuperação do sistema.

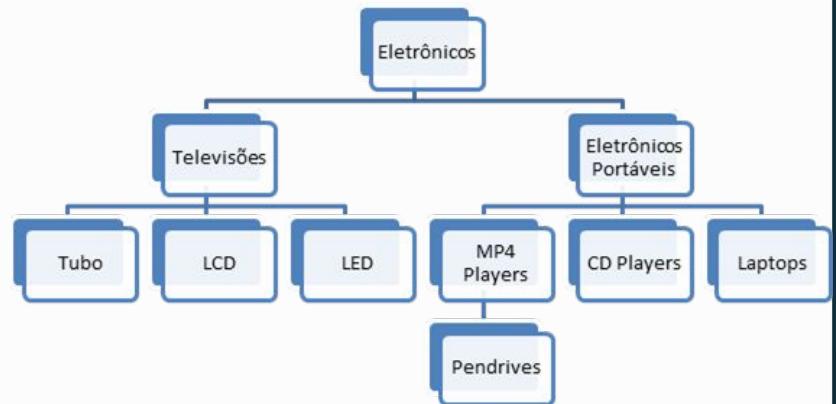
## Durability

Data is persisted after transaction is committed even in a system failure



# Modelos de Dados

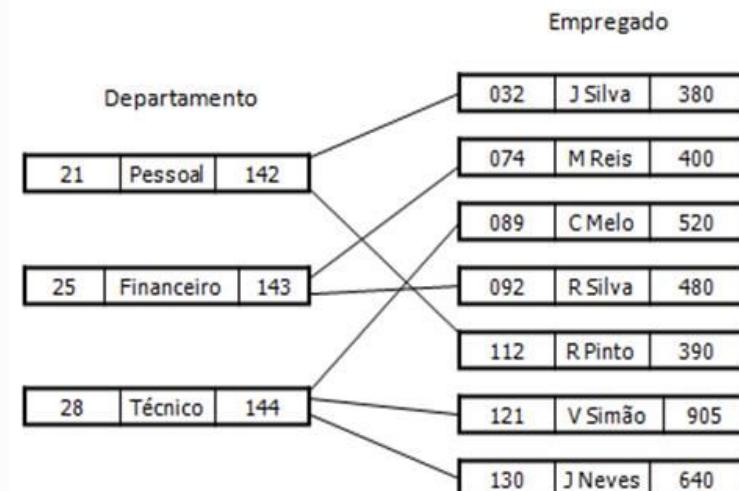
- Os modelos de dados são fundamentais para a estruturação dos sistemas de banco de dados ao longo do tempo.
- Os principais modelos históricos incluem o hierárquico, de redes e relacional, cada um com características e usos distintos.
- **Modelo Hierárquico**
  - Desenvolvido nos anos 1960 com o IMS da IBM, o modelo hierárquico foi eficaz para estruturas de dados previsíveis, mas limitado para consultas complexas e alterações na estrutura.
  - Organiza os dados em uma estrutura de árvore com um único pai e vários filhos, ideal para representar relações hierárquicas naturais.
  - Exemplo prático:
    - Em um sistema de gerenciamento de produtos, categorias como "Eletrônicos" contêm subcategorias como "Computadores," criando uma hierarquia clara.



# Modelos de Dados

- **Modelo em Redes**

- Extensão do Modelo Hierárquico
- Desenvolvido nos anos 1970 pelo CODASYL, o modelo de redes foi uma evolução do hierárquico, oferecendo flexibilidade adicional.
- O modelo de redes permite que um registro tenha vários pais, formando uma estrutura de dados mais flexível que o modelo hierárquico.
- Exemplo prático:
  - Em um sistema de gerenciamento de cursos universitários, alunos podem estar inscritos em vários cursos, e cada curso pode ter vários alunos, criando uma rede de relacionamentos.
- No entanto, a navegação e a consulta aos dados ainda eram complexas, exigindo um entendimento detalhado da estrutura do banco de dados para sua manipulação eficiente.



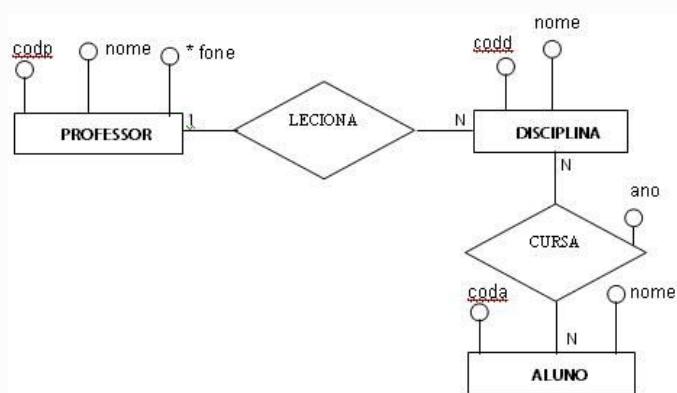
# Modelos de Dados

- **Modelo Relacional**

- Os dados no modelo relacional são organizados em tabelas, onde cada tabela representa uma entidade e as colunas representam atributos.
- Proposto por Edgar F. Codd em 1970, o modelo relacional simplificou o gerenciamento de dados ao abstrair a complexidade dos relacionamentos através de chaves primárias e estrangeiras.
- Exemplo prático:
  - Em um sistema de gerenciamento de biblioteca, a tabela "Livros" possui colunas como "ID do Livro," "Título" e "Autor," e a tabela "Empréstimos" possui colunas como "ID do Empréstimo," "ID do Livro" e "Data de Devolução." A relação entre essas tabelas é estabelecida pelo "ID do Livro."
- A introdução do SQL facilitou consultas complexas e manipulação de dados, impulsionando sua ampla adoção em várias indústrias.

- **Vantagens e Uso do Modelo Relacional**

- Reconhecido pela sua eficiência, robustez e facilidade de uso, o modelo relacional se tornou o padrão dominante em sistemas de banco de dados.
- É ideal para aplicações que requerem manipulação simples e eficiente de dados, promovendo uma gestão organizada e estruturada das informações.





# Estudo de Bancos de Dados Relacionais (SQL)

- **Popularidade e Estabilidade:**
  - Bancos de dados relacionais são amplamente utilizados na indústria devido à sua estabilidade e popularidade duradoura. Com uma base teórica robusta, eles são bem compreendidos e estabelecidos na prática.
- **Transações ACID:**
  - Para aplicações críticas, a garantia de transações ACID (Atomicidade, Consistência, Isolamento e Durabilidade) é crucial. Bancos de dados relacionais são projetados para oferecer suporte robusto a essas propriedades, assegurando a integridade e confiabilidade dos dados.
- **Linguagem SQL:**
  - A Linguagem SQL (Structured Query Language) é padronizada e poderosa para gerenciamento e manipulação de dados.
  - Sua ampla adoção facilita a interoperabilidade entre diferentes sistemas de banco de dados e simplifica o desenvolvimento de consultas complexas.
- **Modelagem de Dados Estruturados:**
  - A modelagem em bancos de dados relacionais promove uma estrutura clara e bem definida, facilitando a compreensão e manutenção da integridade dos dados.
  - Essa abordagem é vantajosa para organizar dados de forma eficiente e escalável.

# Introdução aos Modelos de Bancos de Dados Relacionais

- A primeira coisa a se fazer para desenvolver um sistema que utiliza banco de dados é a compreensão do problema que envolve os seguintes passos:
  - Identificação do Contexto:
    - Entenda qual é o cenário ou contexto em que o sistema será aplicado.
    - Por exemplo, se estamos falando de uma loja, é importante saber se é uma loja física, online, de que tipo de produtos ela vende, qual é o público-alvo, etc.
      - Exemplo: Sistema de biblioteca.
  - Objetivos do Sistema:
    - Determine claramente quais são os objetivos e propósitos principais do sistema. Isso pode incluir facilitar vendas, gerenciar estoques, melhorar o atendimento ao cliente, entre outros.
      - Exemplo: Sistema de biblioteca.
  - Problemas a serem Resolvidos:
    - Identifique quais são os problemas ou desafios que o sistema deve resolver.
    - Por exemplo, se a loja enfrenta dificuldades no controle de estoque ou na gestão de clientes, esses são problemas específicos a serem considerados na análise inicial.
      - Exemplo: Sistema de biblioteca.



# Introdução aos Modelos de Bancos de Dados Relacionais

- **Levantamento de Informações Necessárias:**
  - Identificação das Necessidades dos Usuários:
    - Descubra quem serão os usuários finais do sistema e quais são suas necessidades e expectativas em relação ao sistema.
    - Por exemplo, os funcionários da loja precisam de um sistema que seja fácil de usar e que forneça informações precisas e atualizadas.
      - Exemplo: Biblioteca
- **Entendimento dos Processos Atuais:**
  - Se houver processos existentes que o sistema deve substituir ou melhorar, é crucial entender como esses processos funcionam atualmente.
  - Isso ajuda a identificar lacunas e áreas para melhorias no novo sistema.

**unesp** UNIVERSIDADE ESTADUAL PAULISTA  
Faculdade de Ciências e Letras de Assis

→ **DADOS DO SOLICITANTE** DATA: \_\_\_/\_\_\_/\_\_\_

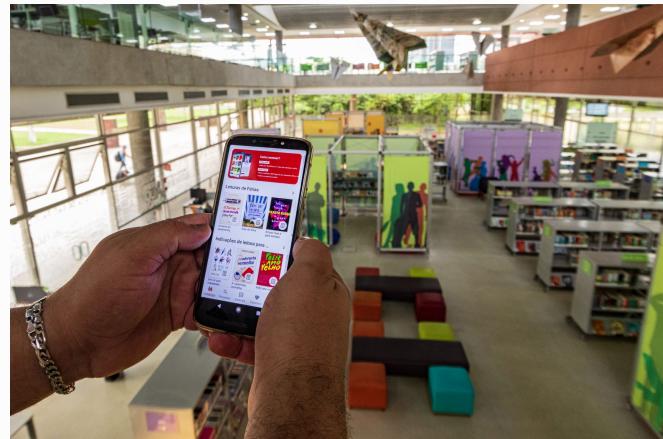
Nome: \_\_\_\_\_  
Categoria:  Graduação  Pós-Graduação  Docente  Funcionário  Ex-aluno  
Curso: \_\_\_\_\_ CPF: \_\_\_\_\_  
E-mail para contato: \_\_\_\_\_

→ **DADOS DA OBRA**

Tipo de material:  Livro  Tese  
Título: \_\_\_\_\_  
Autor: \_\_\_\_\_  
Observações: \_\_\_\_\_

# Introdução aos Modelos de Bancos de Dados Relacionais

- **Identificação das entidades principais envolvidas no sistema.**
  - Entidades são os principais objetos ou conceitos do mundo real que o sistema precisa gerenciar.
  - Por exemplo, em uma loja, entidades podem incluir Clientes, Produtos, Pedidos, etc. Pense nos principais "atores" ou "objetos" que o sistema vai lidar.
    - Exemplo: Biblioteca
- **Identificação dos Atributos das Entidades**
  - Para cada entidade identificada, determine os atributos necessários para descrever completamente cada uma delas.
  - Atributos são características específicas que definem cada entidade.
  - Por exemplo, para a entidade "Cliente", os atributos podem incluir Nome, Endereço, CPF, etc. Para "Produto", pode incluir Descrição, Preço, Tamanho, Cor, etc.
    - Exemplo: Biblioteca



# Introdução aos Modelos de Bancos de Dados Relacionais

- **Exemplo 2: Sistema para o RH de uma empresa de limpeza.**
  - Qual é o cenário ou contexto em que o sistema será aplicado?
  - Quais são os objetivos e propósitos principais do sistema?
  - Quais são os problemas ou desafios que o sistema deve resolver?
  - Quem serão os usuários finais do sistema e quais são suas necessidades e expectativas?
  - Quais são os principais objetos ou conceitos do mundo real (entidades) que o sistema precisa armazenar e gerenciar?
  - Quais os atributos necessários para descrever completamente cada um objetos ou conceitos do mundo real (entidades)?



# Introdução aos Modelos de Bancos de Dados Relacionais

- Exemplo 3: Sistema para uma academia.
- Qual é o cenário ou contexto em que o sistema será aplicado?
  - Quais são os objetivos e propósitos principais do sistema?
  - Quais são os problemas ou desafios que o sistema deve resolver?
  - Quem serão os usuários finais do sistema e quais são suas necessidades e expectativas?
  - Quais são os principais objetos ou conceitos do mundo real (entidades) que o sistema precisa armazenar e gerenciar?
  - Quais os atributos necessários para descrever completamente cada um objetos ou conceitos do mundo real (entidades)?

