

Fundamentos de Bancos de Dados

Capítulo 1. Introdução

Prof. Diego Bernardes

Fundamentos de Bancos de Dados

Aula 1.1. Introdução aos Bancos de Dados

Prof. Diego Bernardes

Nesta aula



- ☐ Introdução aos Bancos de Dados.
- ☐ Definições iniciais e nivelamento de conceitos.

Introdução

SGBD - Sistema Gerenciador de Bancos de Dados:

“Coleção de dados inter-relacionados e um conjunto de programas para acessar esses dados”.

Dados

- Chamamos de Dados fatos conhecidos que podem ser registrados e possuem significado implícito. (ex.: nome, telefone, CPF etc.)
- É a menor unidade de informação que pode ser armazenada em um banco de dados.

Bancos de Dados

- Coleção de dados com informações relevantes que é armazenada em algum local onde possa ser recuperada posteriormente.
Ex.: pastas (de papéis), arquivos (de papéis), hd's, fitas etc.
- Bancos de dados possuem alguma ligação com o mundo real, com a fonte que gera as informações, usuários, algum ator que futuramente possa necessitar das informações armazenadas no banco de dados.

SGBDs X Bancos de Dados

- O Sistema Gerenciador de Bancos de Dados é um sistema que é projetado para gerir os volumes de informações que são armazenados em um banco de dados.
- O SGBD deve fornecer mecanismos de inserção, recuperação, alteração e remoção de todas as informações do banco de dados.
- Exemplos de SGBD's:
Oracle, SQL Server, MySQL, PostgreSQL, DB2 etc.

Finalidade dos SGBDs

Reduzir a redundância e inconsistência nos dados

- Garantir que uma determinada informação esteja armazenada em apenas um local.
- Alterações nos dados deverão ser controladas e automaticamente propagadas à todos os usuários que acessam a informação.

Finalidade dos SGBDs

Reduzir a dificuldade no acesso aos dados

- Os SGBD's fornecem linguagens e mecanismos eficientes para que uma determinada informação seja encontrada.
- Exemplo:
Em um sistema onde informações de alunos estão armazenadas em arquivos, como encontrar todos os alunos que possuem 25 anos?

Finalidade dos SGBDs

Garantir o isolamento dos dados

- Manter todos os dados gravados no banco de dados com o mesmo tipo de formato, bem como impedir que o dados seja acessado de outro local que não seja o SGBD, garantindo, assim, um único ponto de acesso ao dado.

Finalidade dos SGBDs

Minimizar problemas de integridade

- Algumas informações de uma empresa devem seguir algumas regras ou restrições, e o SGBD provê formas mais fáceis de garantir a integridade das informações através dessas restrições.
- Exemplo:
Todas os alunos do IGTI obrigatoriamente devem estar cadastrados em um curso para poder cursar as disciplinas.

Finalidade dos SGBDs

Resolver problemas de atomicidade

- O SGBD deve garantir todas as operações atômicas, ou seja, todas as operações ou transações que devem acontecer completamente ou devem ser desfeitas caso ocorram falhas, são gerenciadas pelo SGBD.

Finalidade dos SGBDs

Resolver problemas de atomicidade

- Exemplo:

Transferir um valor de uma conta corrente a outra. Etapas:

Ler o saldo da conta 01.

Retirar o valor da conta 01.

Ler o valor da conta 02.

Somar o valor retirado da conta 01 à conta 02.

Gravar o novo saldo das duas contas.

Caso um dos passos anterior falhe, todos os passos devem ser desfeitos

Finalidade dos SGBDs

Problemas de segurança

- Os SGBD's possuem usuários e perfis de acesso, de modo que nem todos os usuários tem acesso a todas as informações.
- Exemplo:
Apenas os funcionários de gerencia devem ter acesso às informações de folha de pagamento, portanto deve-se criar um perfil diferenciado para esses usuários, impedindo que usuários não autorizados consigam ler essa informação.

Abstração de Dados

- Muitos usuários de uma corporação possuem a necessidade de acessar e compreender como os dados estão armazenados e distribuídos dentro do banco de dados, mas muitos deles não são especialistas em computação.
- Existem, portanto, maneiras diferentes de abstrair os dados, de forma que todos os usuários possam compreender o banco de dados.

Abstração de Dados

- **Nível Conceitual**

Nível mais alto, descreve apenas parte do banco de dados, exibe apenas a parte do sistema que o usuário necessita ver, de forma simplificada.

- **Nível Lógico**

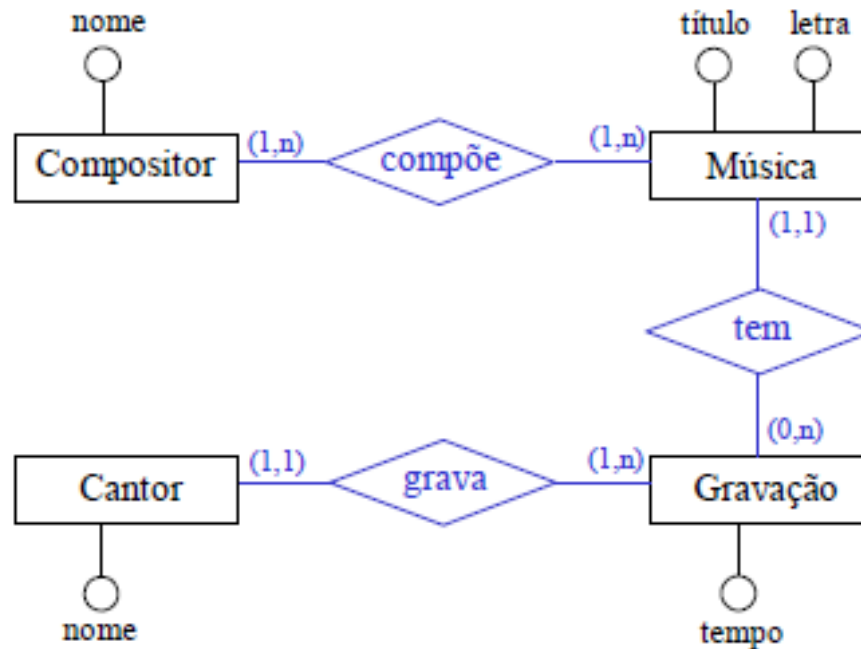
O nível lógico é o nível intermediário, neste nível é possível representar todo o banco de dados, com suas estruturas e relacionamentos, mas sem a preocupação de como o dado será gravado no disco.

- **Nível Físico**

Nível de abstração mais baixo, ou seja, demonstra como os dados serão armazenados fisicamente no disco.

Modelo de Dados

- O modelo de dados é a forma pela qual descrevemos o projeto de bancos de dados. O modelo de dados representa todos os dados, a forma como os dados se relacionam e as possíveis restrições sobre os mesmos.



Conclusão

- ✓ Estabelecemos os conceitos que serão discutidos ao longo do módulo.
- ✓ Discutimos o papel e a importância dos SGBDs.

Próxima aula...



- ❑ Introdução à Modelagem de Dados.



Fundamentos de Bancos de Dados

Aula 1.2. Introdução à Modelagem de Dados

Prof. Diego Bernardes

Nesta aula



- ❑ Introdução à Modelagem de Dados.

Modelagem Entidade-Relacionamento



- Elementos:
 - Entidades
 - Conjuntos de “elementos” que possuem características próprias.
 - Atributos
 - Representam as características de uma Entidade.
 - Relacionamentos
 - Vínculos ou associações entre Entidades.



Entidade

- Conjunto de objetos sobre os quais é preciso armazenar informações.
- Conjunto de vários elementos (mais que 1).
- Conjuntos de elementos distinguíveis que aceitam um código para diferenciá-los.
- Seus atributos NÃO dependem de outras entidades.
- Exemplos: de possíveis Entidades:
pessoas, locais, objetos, documentos etc.

- Entidade



Funcionários

matrícula nome endereço

Notas Fiscais

série número data
emissão

Produtos

código descrição unidade

Cargos

código descrição pré-requisitos

Atributos

- Informações úteis a respeito de uma entidade ou relacionamento.
- Os atributos de uma entidade permanecem constantes para todos os seus relacionamentos.
- Os atributos de uma entidade são independentes de todas as demais entidades.

Atributos

- Chave:
 - seu valor representa um elemento da entidade.
 - seu valor é único para a entidade.
 - deve ser sublinhado.
- Composto:
 - necessita ser dividido em sub-atributos, para que seu significado seja melhor compreendido.
- Multi-valorado:
 - Pode assumir mais do que um valor para cada entidade.

Atributos



Funcionários

matrícula nome endereço

Notas Fiscais

série número data
emissão

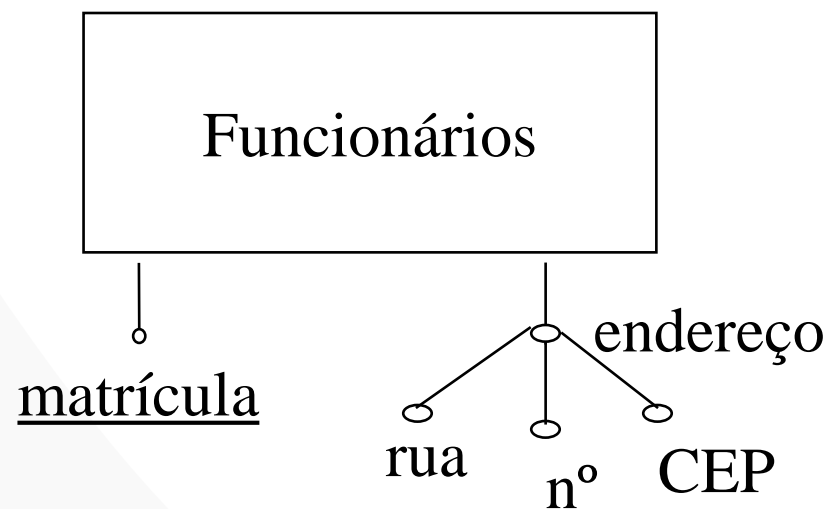
Produtos

código descrição unidade

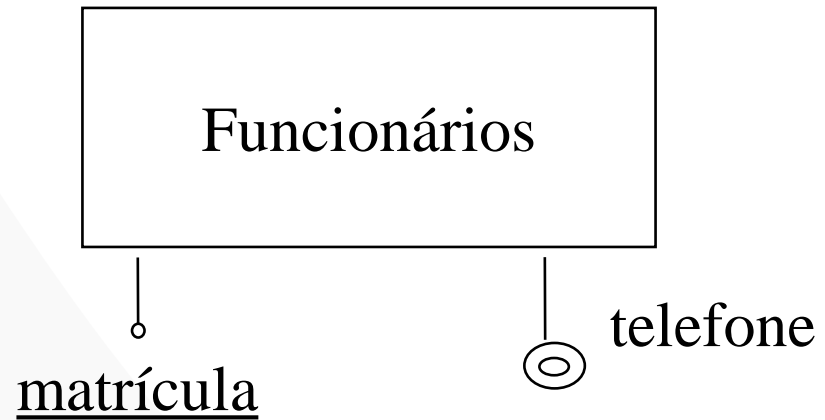
Cargos

código descrição pré-requisitos

Atributos Compostos

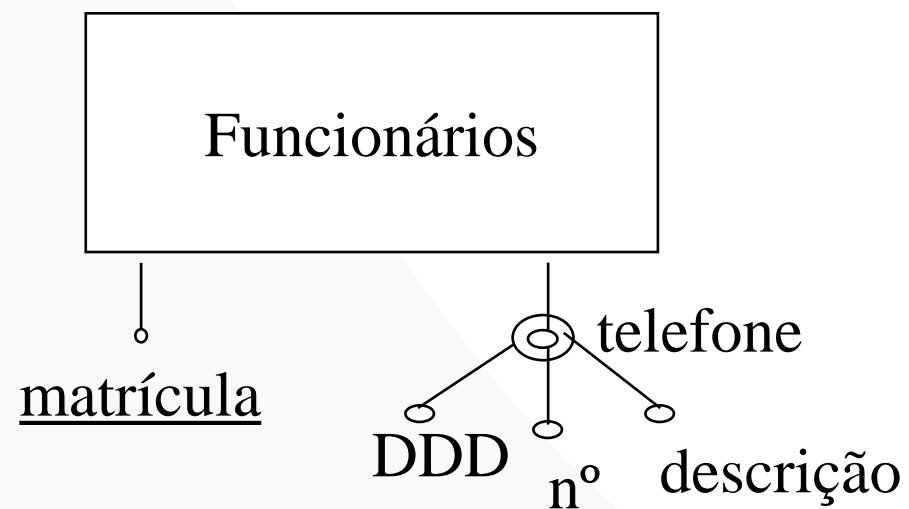


Atributos Multivalorados

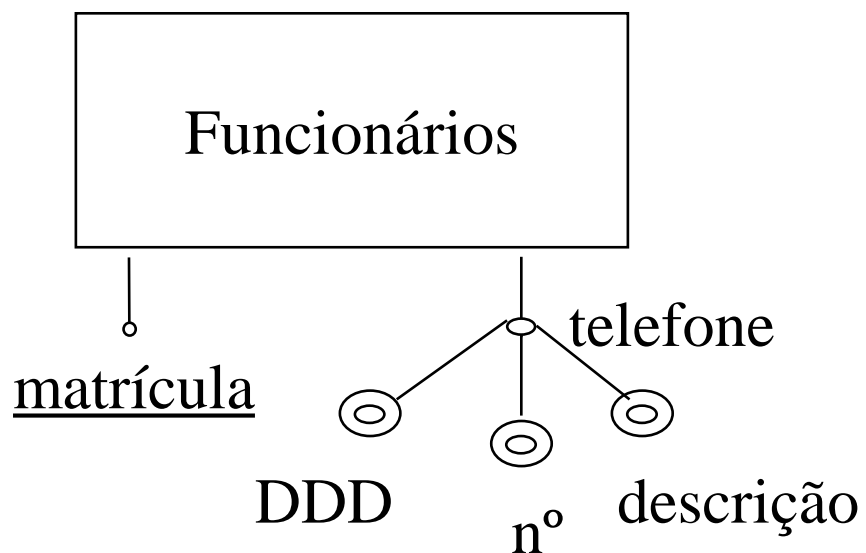


Qual solução adotar?

Caso 1

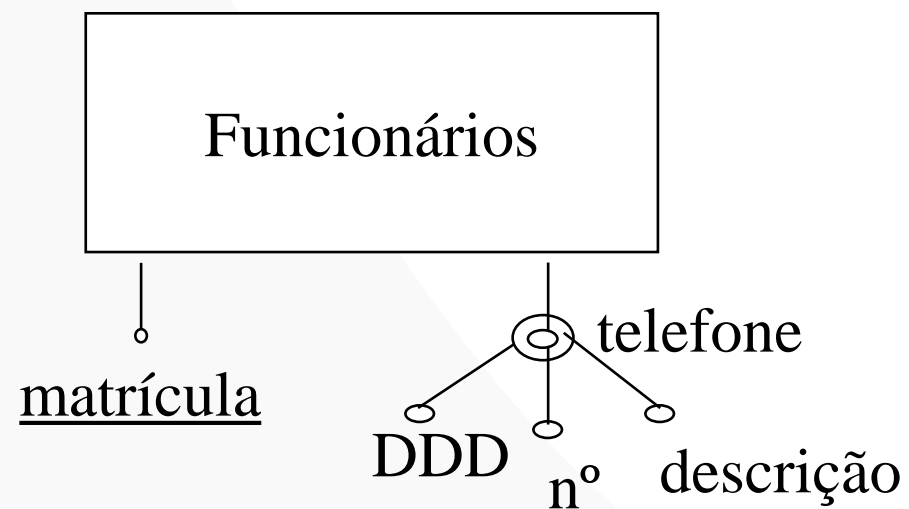


Caso 2

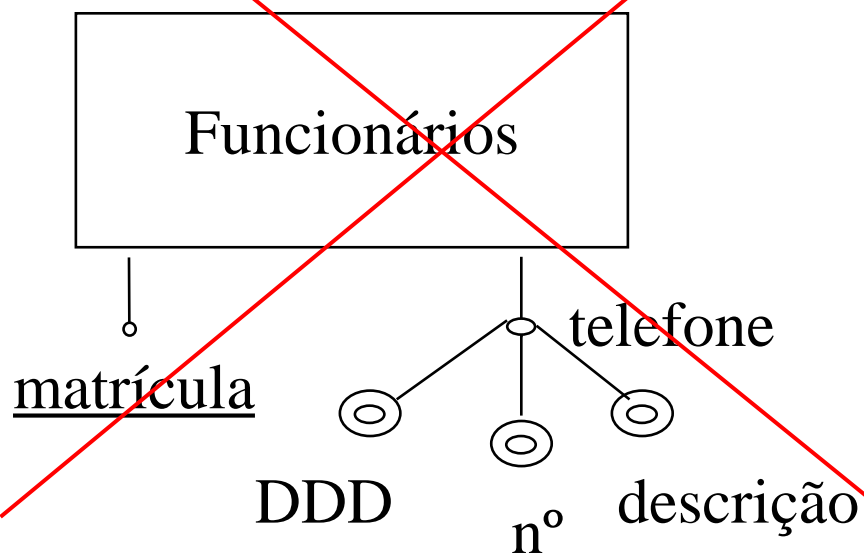


Qual solução adotar?

Caso 1



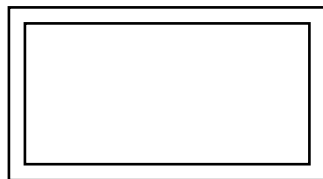
Caso 2



Entidades Fracas

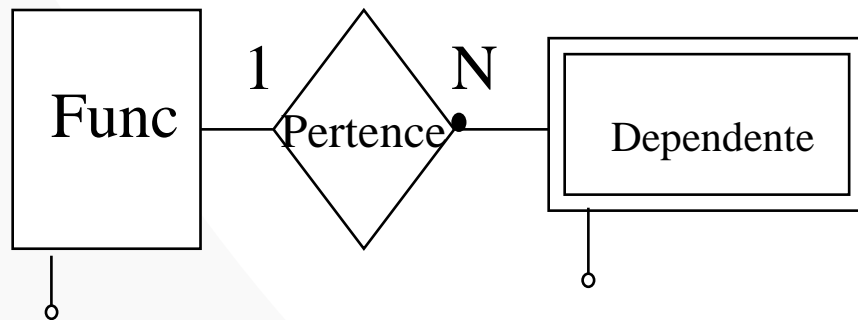
- Dependem de uma “entidade forte”.
- A Entidade Fraca é representada por:
- Dependência de Existência.
- Dependência de Identificador.

Representação:



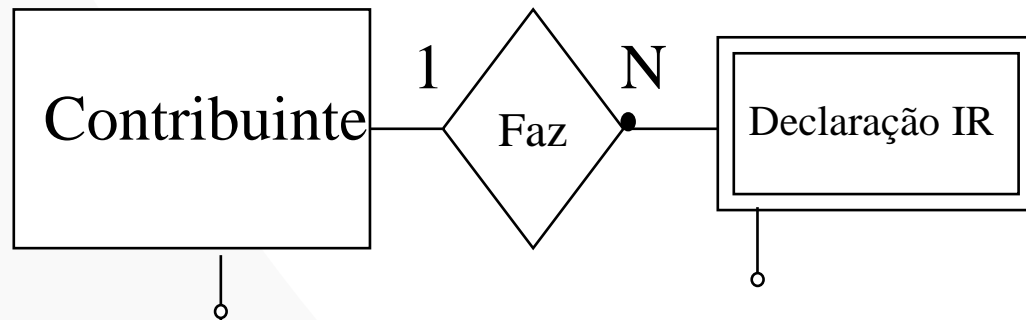
Entidades Fracas

- Necessária Dependência de Existência.



Entidades Fracas

- Necessária Dependência de Identificador.



Próxima aula...



- ☐ Introdução à Modelagem de Dados – Relacionamentos entre entidades.

Fundamentos de Bancos de Dados

Aula 1.3.1. Modelagem de Dados – Relacionamentos (Parte 1)

Prof. Diego Bernardes

Nesta aula



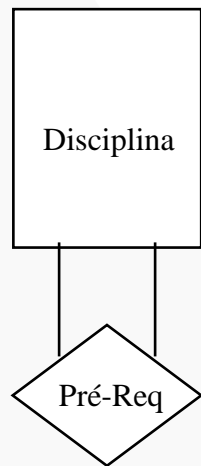
- ❑ Modelagem de Dados – Relacionamentos.

Modelagem ER - Relacionamentos

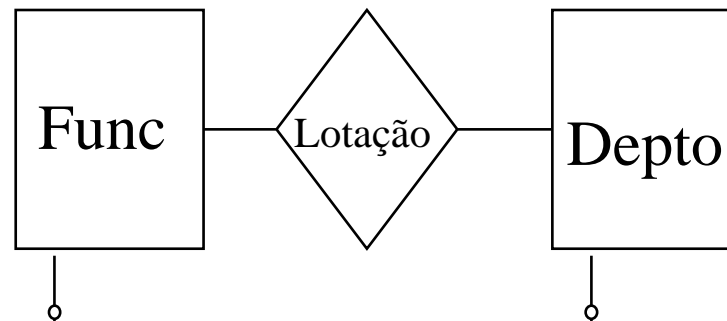
- Associação entre entidades.
- Representam os vínculos que existem entre as entidades no mundo real.
- São representados por losangos.
- Ex.: Em um sistema de controle acadêmico o relacionamento MATRÍCULA, vincula um ALUNO a uma DISCIPLINA.

Modelagem ER - Relacionamentos

- Grau de Relacionamento: É igual a quantidade de entidades vinculadas através do relacionamento.



Grau 1



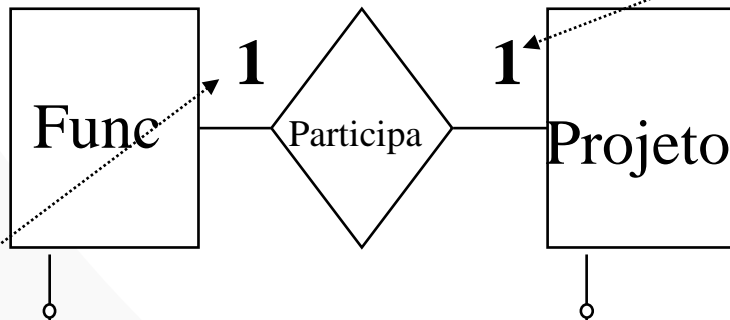
Grau 2

Modelagem ER - Relacionamentos

- Classe:
 - Identifica a quantas vezes cada instância de uma entidade pode participar do relacionamento.
- Para relacionamentos binários temos classes:
 - 1:1
 - 1:N
 - N:N

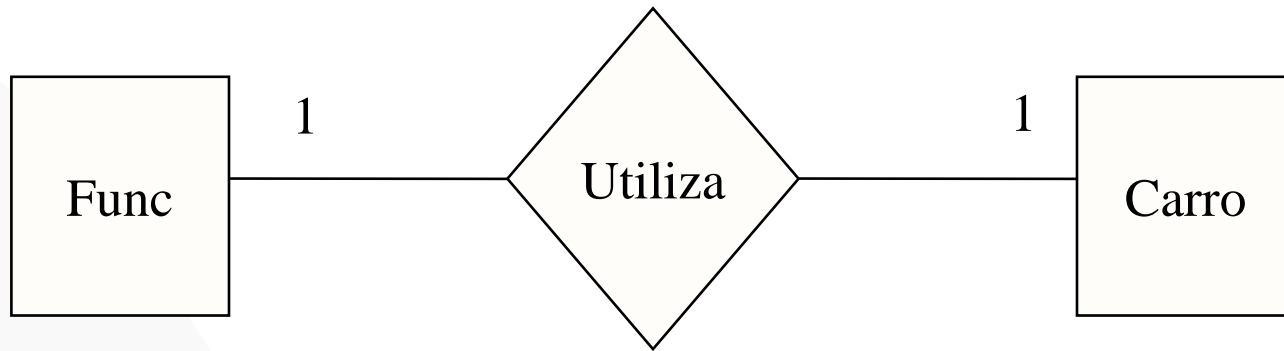
Modelagem ER - Relacionamentos

Cada FUNC participa de quantos PROJETOS?



Cada PROJETO tem a participação de quantos FUNC?

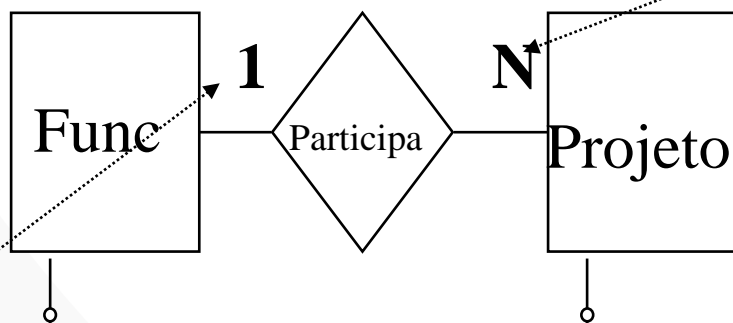
Classe 1:1



Kurt	Fargo
Brian	Mustang
Tonya	Ranger
Scott	Jeep
Nancy	Prizm

Modelagem ER - Relacionamentos

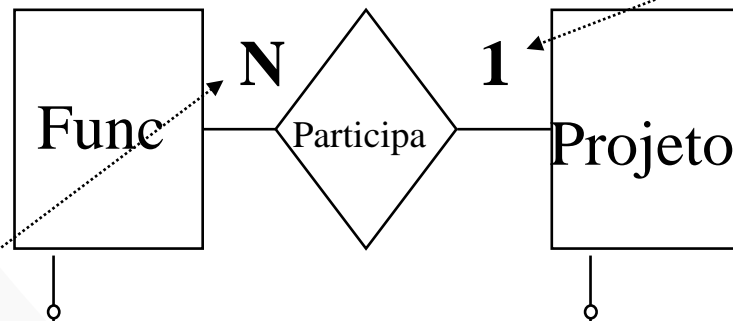
Cada FUNC participa de quantos PROJETOS?



Cada PROJETO tem a participação de quantos FUNC?

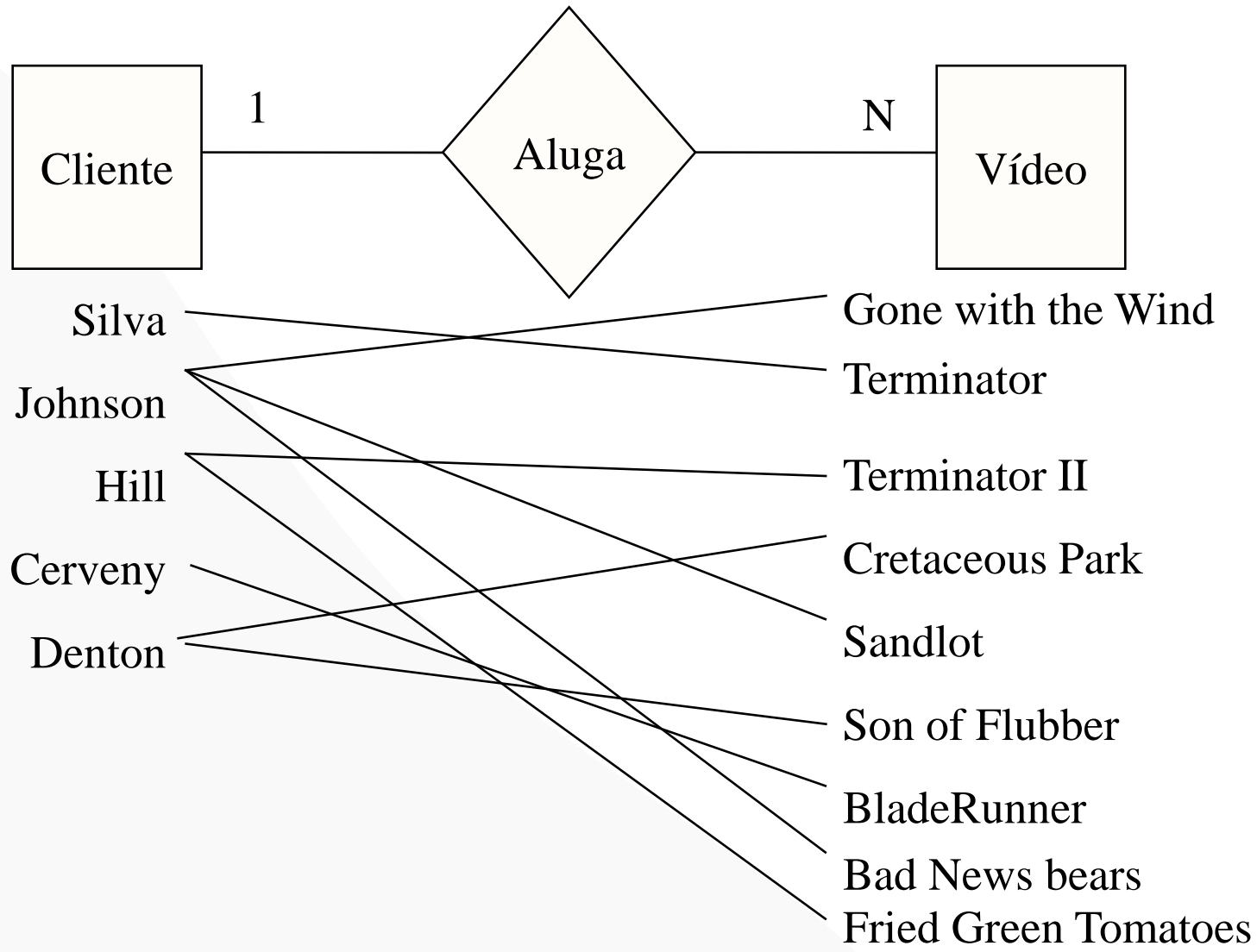
Modelagem ER - Relacionamentos

Cada FUNC participa de quantos PROJETOS?



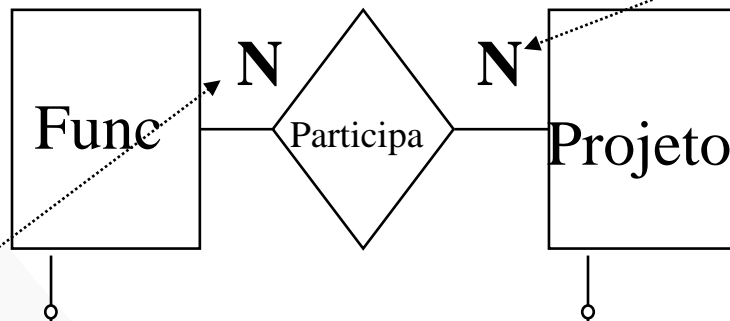
Cada PROJETO tem a participação de quantos FUNC?

Classe 1:N



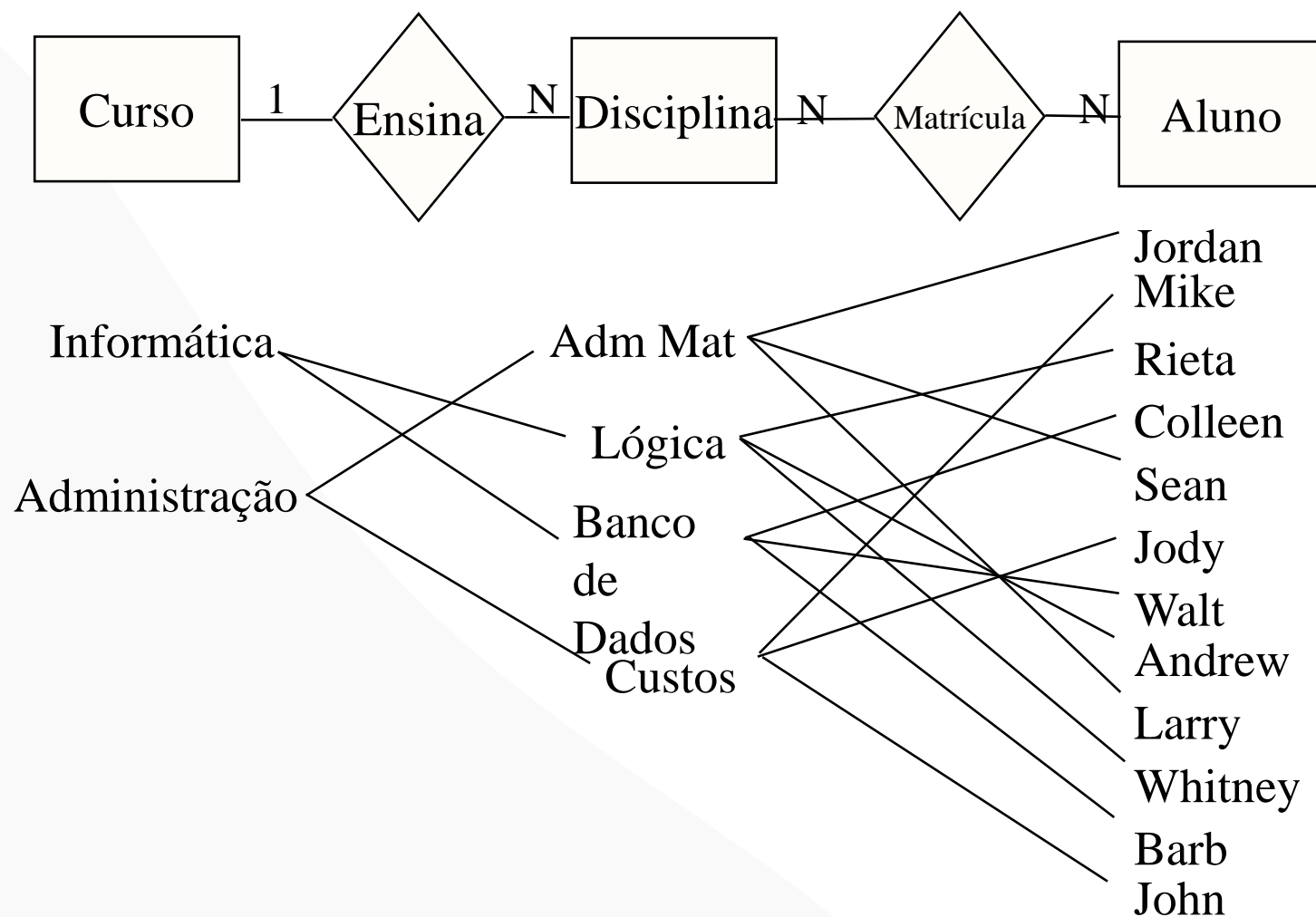
Modelagem ER - Relacionamentos

Cada FUNC participa de quantos PROJETOS?



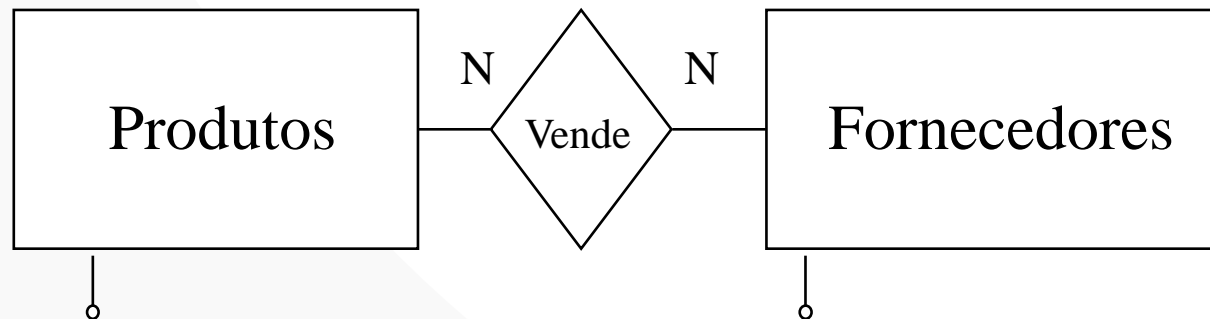
Cada PROJETO tem a participação de quantos FUNC?

Classes 1:N e N:N



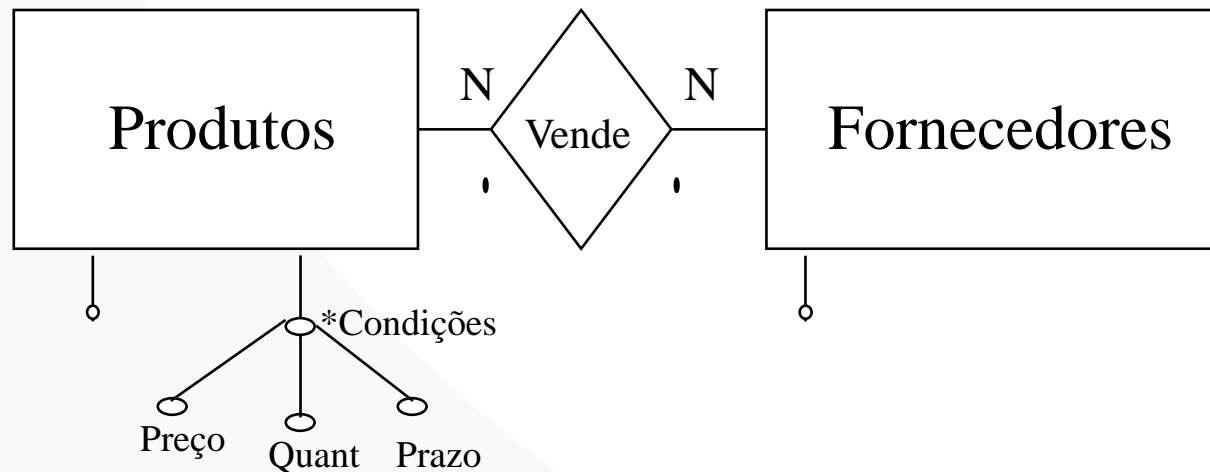
Atributos de Relacionamentos

- A quem pertence os ATRIBUTOS: PREÇO, QUANTIDADE e PRAZO?



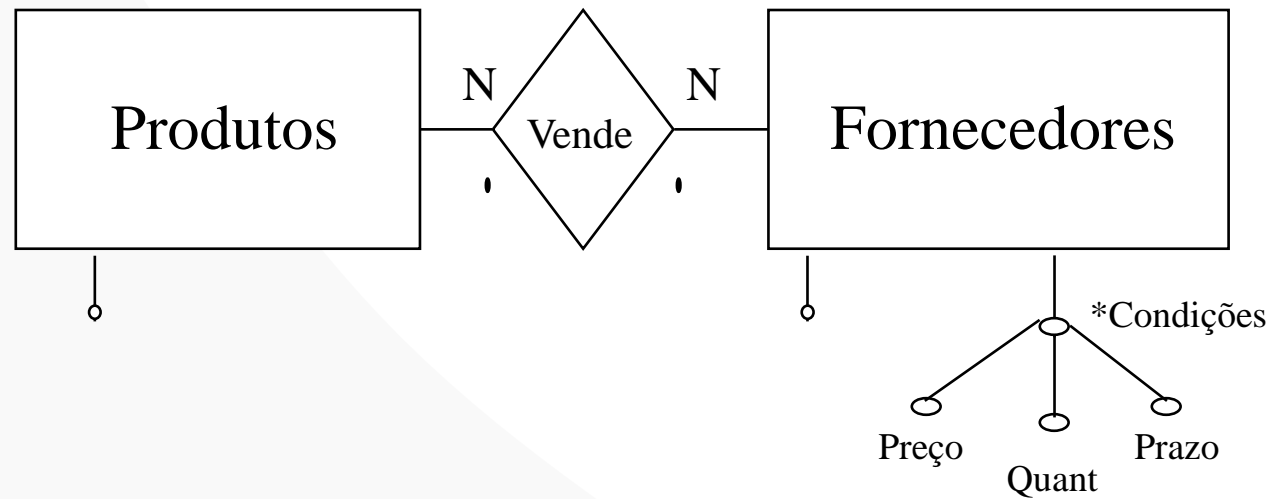
Atributos de Relacionamentos

- PREÇO, QUANTIDADE e PRAZO, não podem pertencer a PRODUTOS, pois se fosse assim TODOS os FORNECEDORES deveriam praticar o mesmo preço.



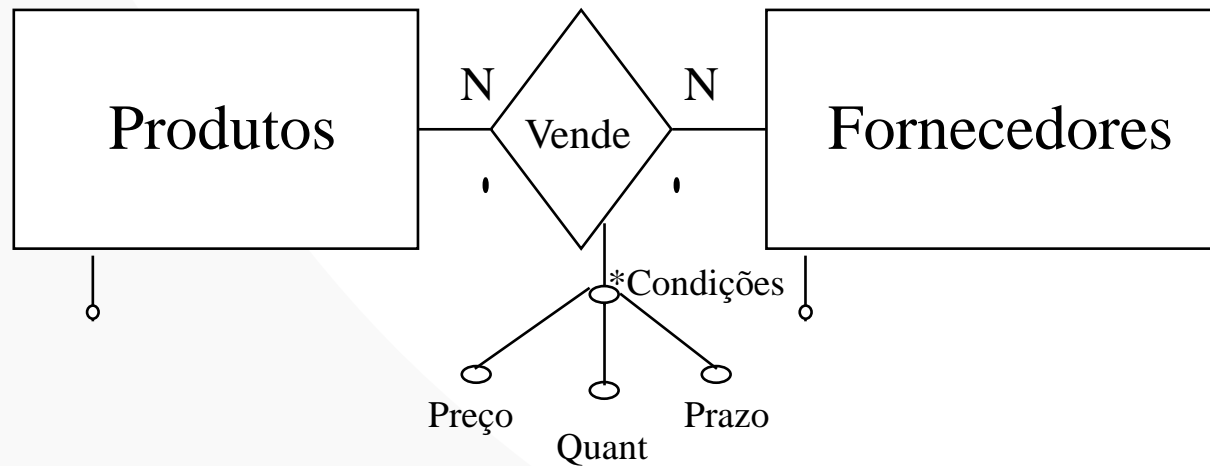
Atributos de Relacionamentos

- PREÇO, QUANTIDADE e PRAZO, não podem pertencer a FORNECEDORES, pois se fosse assim TODOS os PRODUTOS de um fornecedor teriam o mesmo preço.



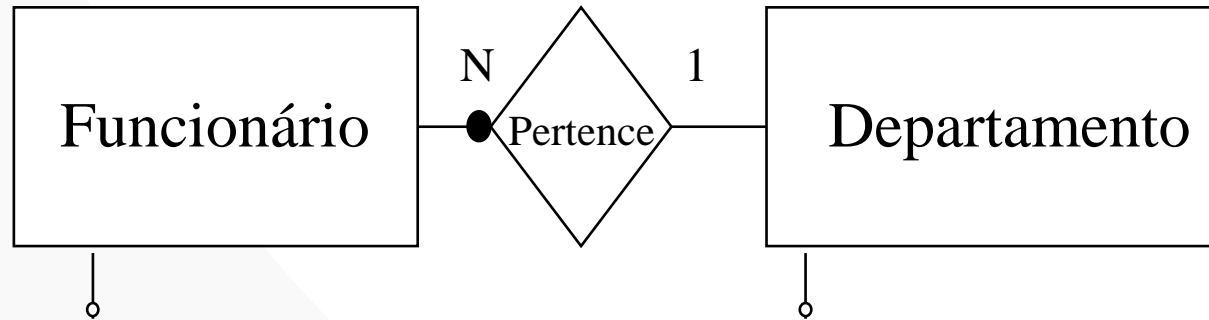
Atributos de Relacionamentos

- Não pertencendo nem a PRODUTOS ou a FORNECEDORES, e sendo relevante no relacionamento VENDA, são atributos do relacionamento.



Atributos de Relacionamentos

- Onde colocar os atributos “Data de Admissão” e “Data de Lotação”?



Próxima aula...



- ❑ Implementação de Modelo de Dados Conceitual.

Fundamentos de Bancos de Dados

Aula 1.3.2. Modelagem de Dados – Relacionamentos (Parte 2)

Prof. Diego Bernardes

Nesta aula



- ❑ Modelagem de Dados – Relacionamentos.

Modelagem ER - Demonstração

- Deseja-se construir um sistema para gestão de recursos humanos de uma empresa.
- Sabe-se que é importante manter o cadastro dos funcionários, quais os respectivos departamentos e sua evolução dentro da empresa.
- Cada departamento tem uma cidade onde está localizado, e consequentemente uma regional, que por sua vez está localizada em um dos países.
- Para controle territorial, é necessário definir qual continente cada filial está localizada.
- Deve-se manter o histórico de todos os cargos que cada funcionário já ocupou.

Modelagem ER - Demonstração

- Informações importantes que devem ser armazenadas:
 - Nome, sobrenome, e-mail e data de admissão dos funcionários;
 - Qual é o gerente de cada funcionário;
 - Qual é o cargo atual de cada funcionário;
 - Quais cargos cada funcionário já ocupou;
 - Qual é o departamento que cada funcionário está vinculado;
 - Qual localidade cada departamento se encontra, bem como seu endereço;
 - Países e continentes de cada departamento.

Próxima aula...



- ❑ Modelo Lógico de Dados.

Fundamentos de Bancos de Dados

Capítulo 2. Modelagem de Dados Relacionais

Prof. Diego Bernardes

Fundamentos de Bancos de Dados

Aula 2.1.1. Modelo Lógico (Parte 1)

Prof. Diego Bernardes

Nesta aula



- ❑ Modelagem de Dados – Modelo Lógico.

Modelagem de Dados

- As formas de modelagem ER e Relacional permitem estabelecer três tipos de modelos:
 - Conceitual: Elaborado para entendimento do negócio, levantamento de requisitos e apresentação aos clientes de negócio.
 - Lógico: Elaborado para equipe de desenvolvimento, projeto de como o banco de dados será estruturado.
 - Físico: Implementação do projeto lógico.
- Projeto Lógico (Modelo Lógico):
 - Obtido a partir da transformação do modelo ER para o Relacional.

Etapas do Projeto de BD

- Verificar Requisitos.
- Obter o modelo conceitual.
- Definir a abordagem de banco de dados a ser utilizada (relacional, orientada a objetos, objeto-relacional).
- Aplicar as regras de derivação específicas.
- Implementar as estruturas no SGBD.

Modelo Lógico

Um modelo de dados lógico é composto por:

- Tabelas ou relações;
- Chaves primária ;
- Chaves estrangeiras.

Modelo Lógico

Tabela Empregado

CodEmp	Nome	CodDeppto	CategFuncional
E5	Souza	D1	C5
E3	Santos	D2	C5
E2	Silva	D1	C2
E1	Soares	D1	C6

*Linha ou
Tupla ou
Registro*

chave primária

chave estrangeira

Modelo Conceitual x Modelo Lógico



Modelo ER	Modelo Relacional
Entidade	Tabela (Relação)
Instância de Entidade	Linha (Tupla)
Atributo	Coluna (Campo)
Atributo Multivalorado	Tabela Auxiliar
Atributo Identificador	Chave
Atributo Composto	Várias Colunas
Relacionamento	Ligações

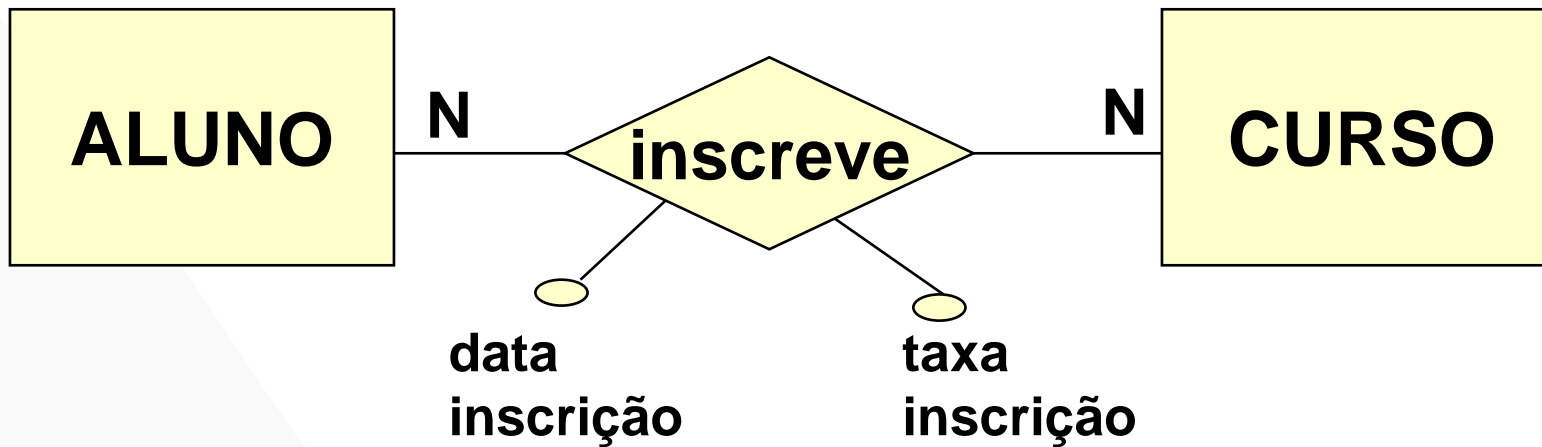
Conversão de ER para Lógico

- A tradução do relacionamento depende da cardinalidade das entidades que participam do mesmo.
- Formas básicas de tradução:
 - Tabela própria para o relacionamento.
 - Colunas adicionais dentro da tabela de entidade.
 - Fusão das entidades em uma.

Relacionamentos NxN

- Para o relacionamento entre A e B de N:N, temos a regra:
 - Criar sempre uma tabela C, agregando as chaves estrangeiras de A e B para formar a chave primária da tabela C, referente ao relacionamento.
 - Caso existam atributos, alocá-los à tabela C como campos normais (descritivos).

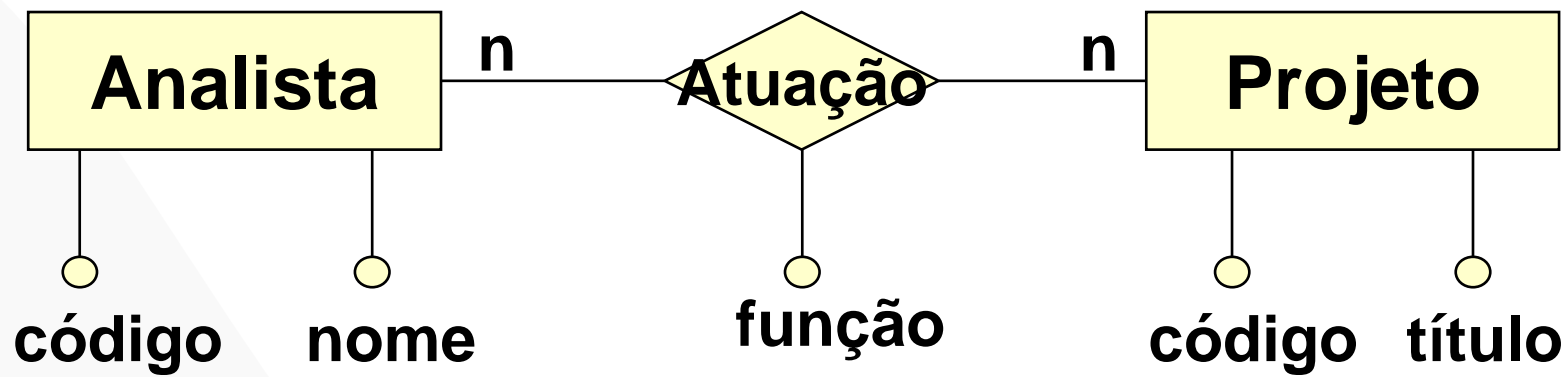
Relacionamentos NxN



Solução:

Criar a tabela INSCREVE contendo seus atributos originais e recebendo as chaves primárias das tabelas CURSO e ALUNO (como chaves estrangeiras).

Relacionamentos NxN



Esquema relacional correspondente:

Analista(codanalista, nome);

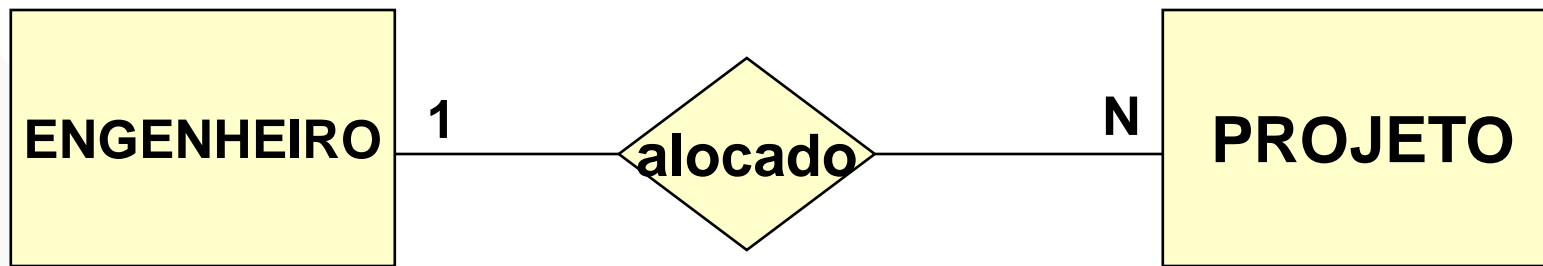
Projeto(codproj, título);

Atuação(codanalista, codproj, função).

Relacionamentos 1xN

- Dado um relacionamento 1:N entre A e B, temos a regra:
- Acrescentar a chave primária da tabela A como chave estrangeira na tabela B (lado N).

Relacionamentos 1xN



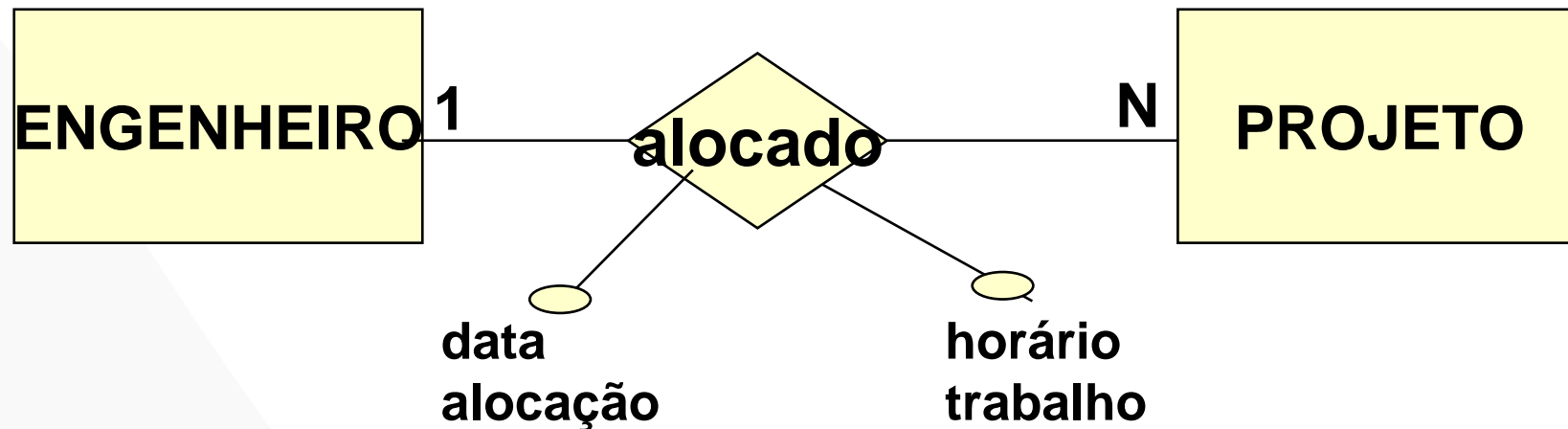
Solução:

Migrar a chave primária da tabela **ENGENHEIRO** para a tabela **PROJETO** (como chave estrangeira).

Relacionamentos 1xN

- Dado um relacionamento 1:N entre A e B, temos as opções:
 - Migrar os atributos do relacionamento para a tabela B (lado N) = > mais comum.
 - Criar uma tabela C para conter as chaves estrangeiras de A e B e alocar os atributos do relacionamento => pouco utilizada.

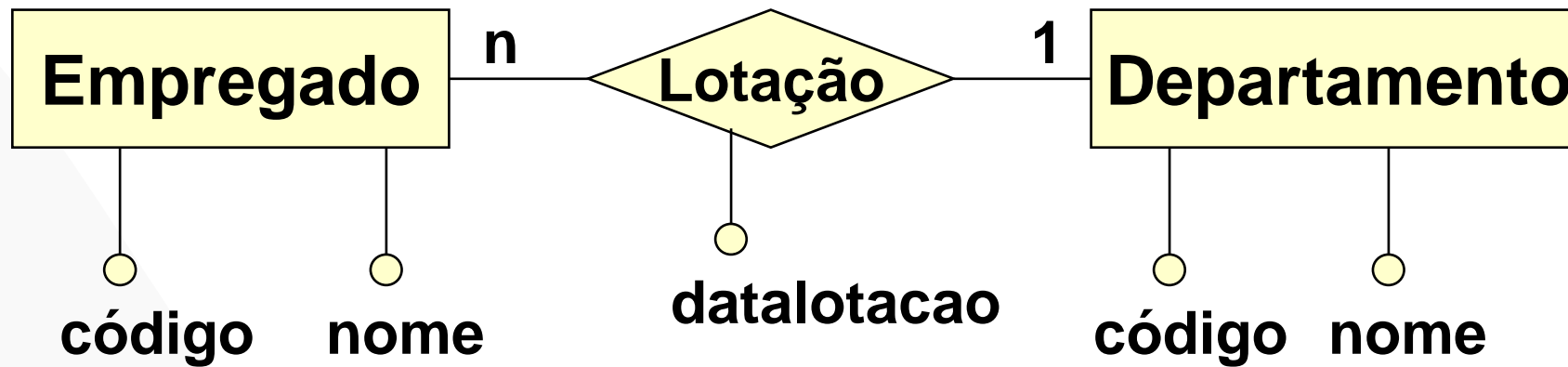
Relacionamentos 1xN



Solução:

Alocar os atributos do relacionamento “alocado” para a tabela PROJETO e migrar a chave primária da tabela ENGENHEIRO para a tabela PROJETO.

Relacionamentos 1xN



Esquema relacional correspondente:

Departamento(coddepto, nome);

Empregado(codemp, nome, **coddepto**, datalotacao).

Relacionamentos 1xN

Empregado:

<u>codemp</u>	nome	<u>coddepto</u>	<u>data lotacao</u>
101	João	1	30/12/1976
102	José	2	12/06/2001
103	Maria	1	21/03/1987

Departamento:

<u>coddepto</u>	nome
1	Gerência
2	Vendas
3	Compras

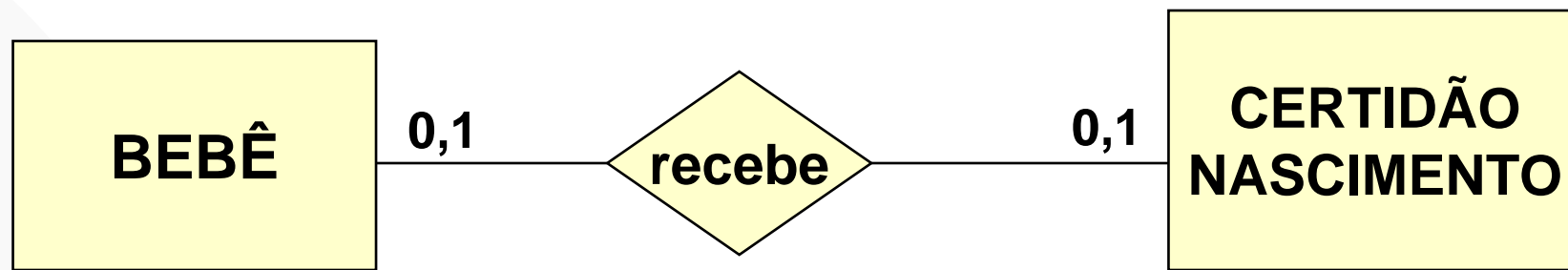
Relacionamentos 1x1

- Dadas 2 entidades A e B e um relacionamento de 1:1, temos as opções:
 - Acrescentar a chave primária da tabela A como chave estrangeira da tabela B.
 - Acrescentar a chave primária da tabela B como chave estrangeira da tabela A.

Relacionamentos 1x1

- Critérios para Escolha:
 - 1º Ver qual tabela nasce antes:
 - Se A surge primeiro, então, migrar a chave estrangeira de A para B.
 - 2º Analisar qual entidade será mais manipulada, a nível de acesso:
 - Se a tabela A será mais manipulada, colocar a chave estrangeira de B nela.
 - 3º Para desempate, observar qual a maior chave (em termos de tamanho):
 - Deverá ser migrada a menor.

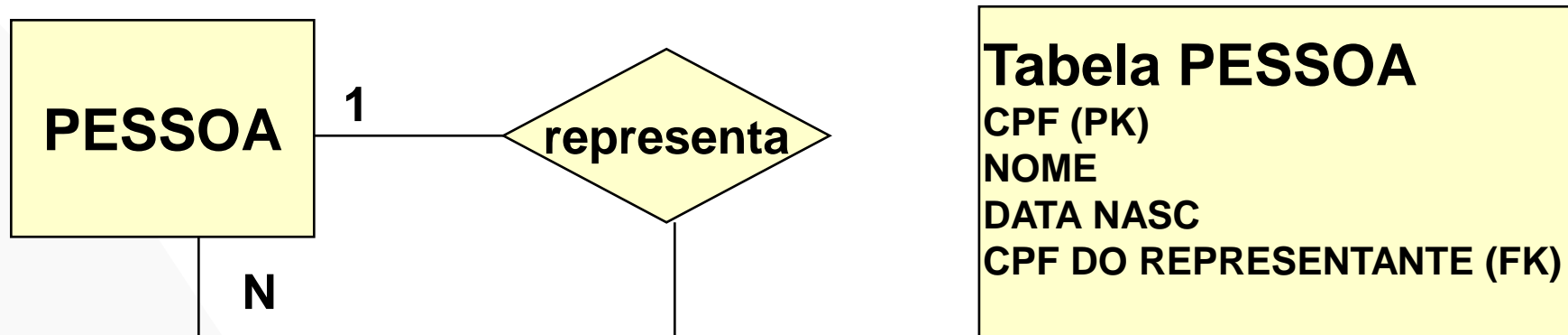
Relacionamentos 1x1



Solução:

Migrar a chave primária de BEBÊ para a tabela CERTIDÃO NASCIMENTO.

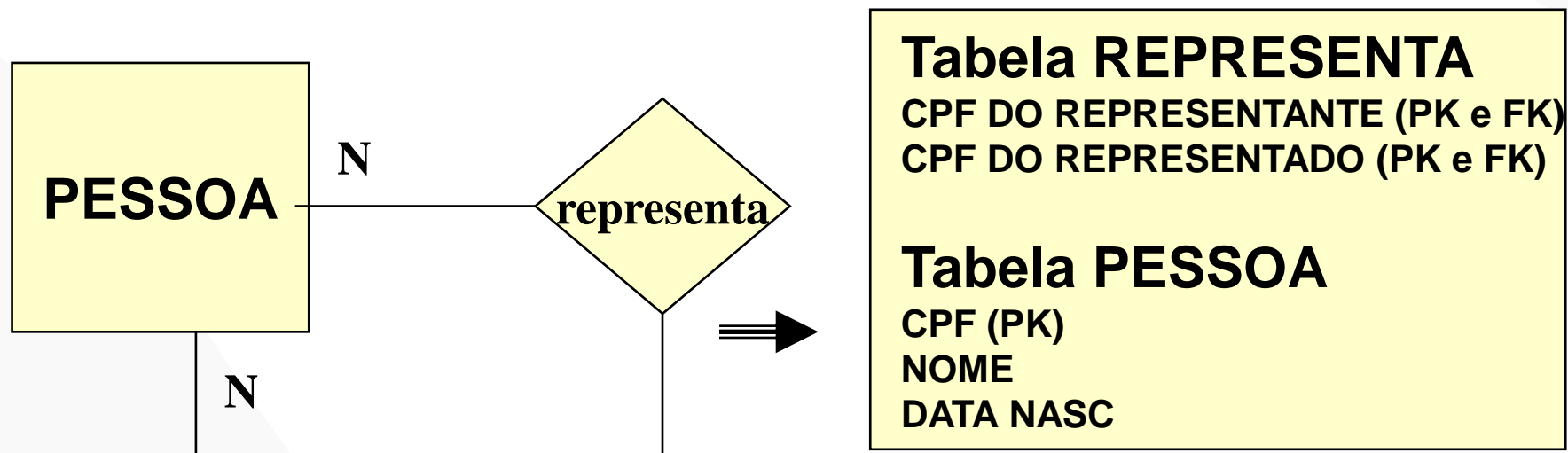
Auto Relacionamento



Solução:

Migrar a chave primária de PESSOA para PESSOA (como chave estrangeira).

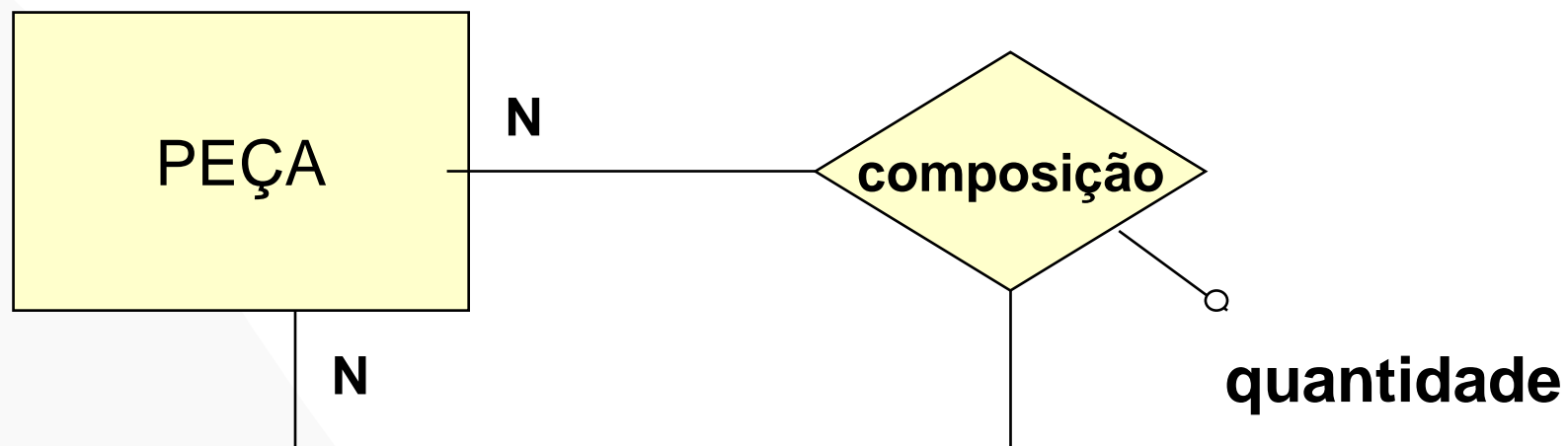
Auto Relacionamento



Solução:

Criar a tabela REPRESENTA com as chaves primárias de PESSOA (estrangeiras que vão formar a chave primária de REPRESENTA).

Auto Relacionamento



Esquema Relacional:

Peca(cod_peca, descrição, peso, cor);

Composição(cod_peca, cod_peca_compo, quantidade).

Próxima aula...



- ❑ Modelo Lógico de Dados.

Fundamentos de Bancos de Dados

Aula 2.1.2. Modelo Lógico (Parte 2)

Prof. Diego Bernardes

Nesta aula



- ❑ Modelo Lógico.

Modelagem Lógica - Demonstração



- .

| Próxima aula...



- ❑ Modelo Lógico de Dados.

Fundamentos de Bancos de Dados

Capítulo 3. Bancos de Dados Não Relacionais

Prof. Diego Bernardes

Fundamentos de Bancos de Dados

Aula 3.1. Introdução aos Bancos de Dados NoSQL

Prof. Diego Bernardes

Nesta aula



- ❑ Introdução aos Bancos de Dados NoSQL.

Introdução ao NoSQL

- **Not Only SQL**
 - Expressão utilizada para definir essa categoria de bancos de dados que não utilizam modelos de dados relacionais.
 - Embora não sejam bancos de dados relacionais, possuem também mecanismos de buscas que recebem instruções SQL.
 - Possuem, também, outros mecanismos de busca além do SQL.

Motivação

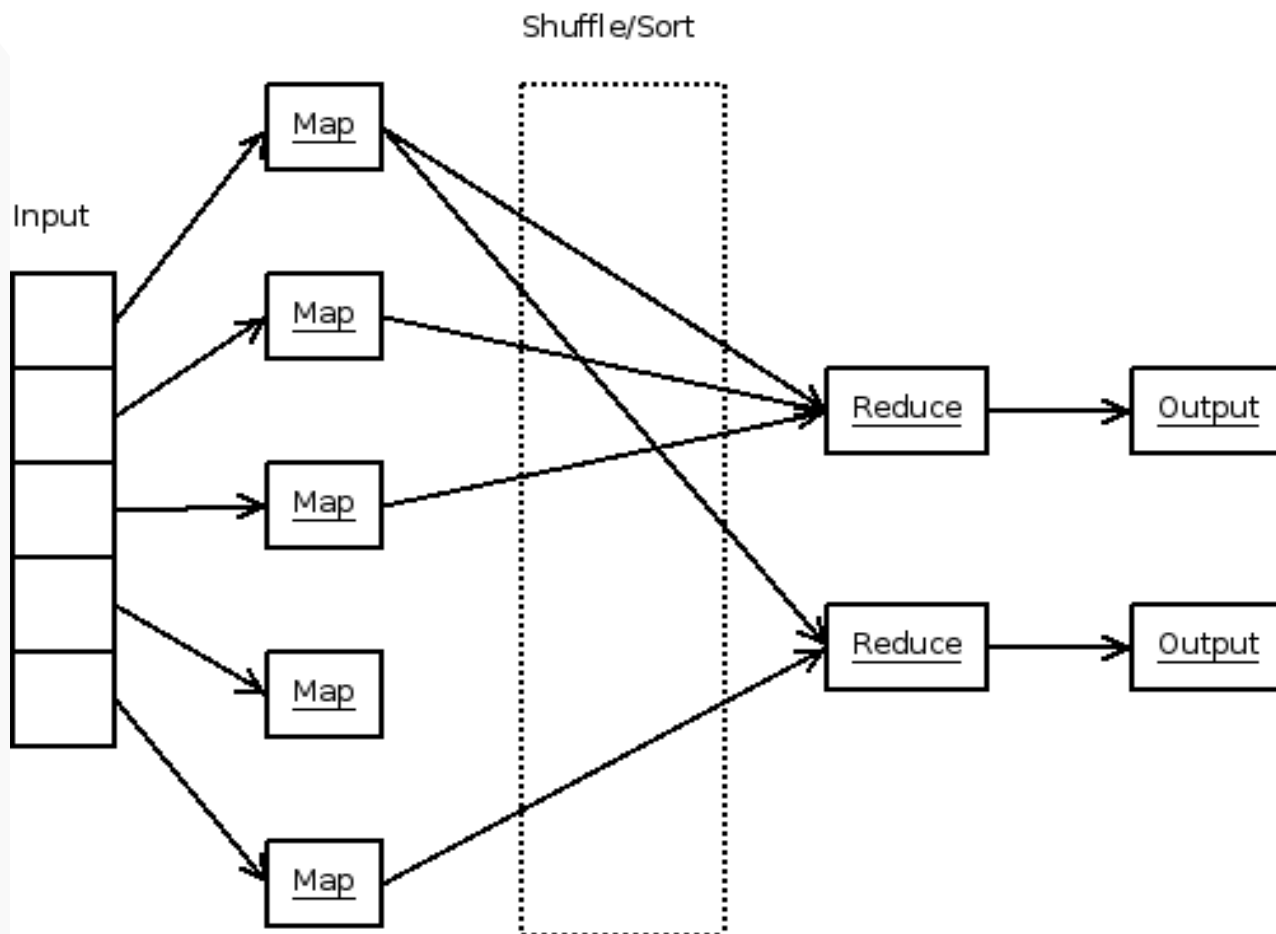
- Crescimento exponencial da geração e necessidade de consumo aos dados.
- Bancos de dados relacionais seriam capazes de manipular grandes volumes de dados?
- Dificuldade de escalabilidade dos Bancos de Dados Relacionais.
 - Bancos de dados relacionais escalam, mas quanto maior o tamanho, mais custoso se torna essa escalabilidade, seja pelo custo de novas máquinas, seja pelo aumento de especialistas nos bancos de dados utilizados.

Características

- Escalabilidade:
 - Crescimento horizontal simplificado.
- Não utilizam esquema/restrições referenciais:
 - Maior flexibilidade.
- Processamento Distribuído, de baixo custo.
- Suporte a replicação.

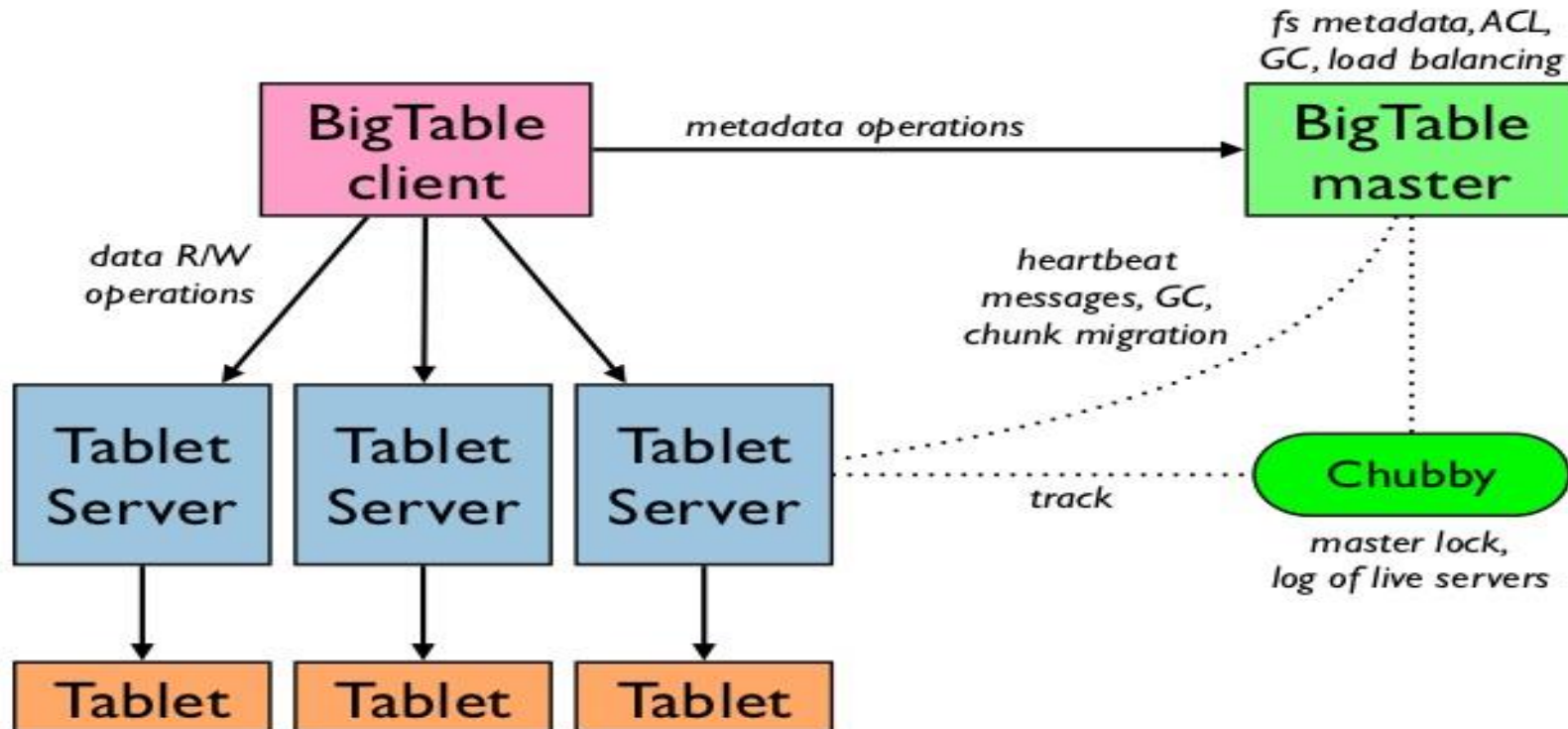
Características

- MAP/REDUCE.



Características

Google BigTable: Architecture



Características

- Bancos de Dados Relacionais utilizam propriedades ACID.
 - Atomicidade;
 - Consistência;
 - Isolamento;
 - Durabilidade.
- Bancos NoSQL utilizam propriedades BASE.
 - Basicamente disponível;
 - Estado leve;
 - Eventualmente consistente.

ACID x BASE



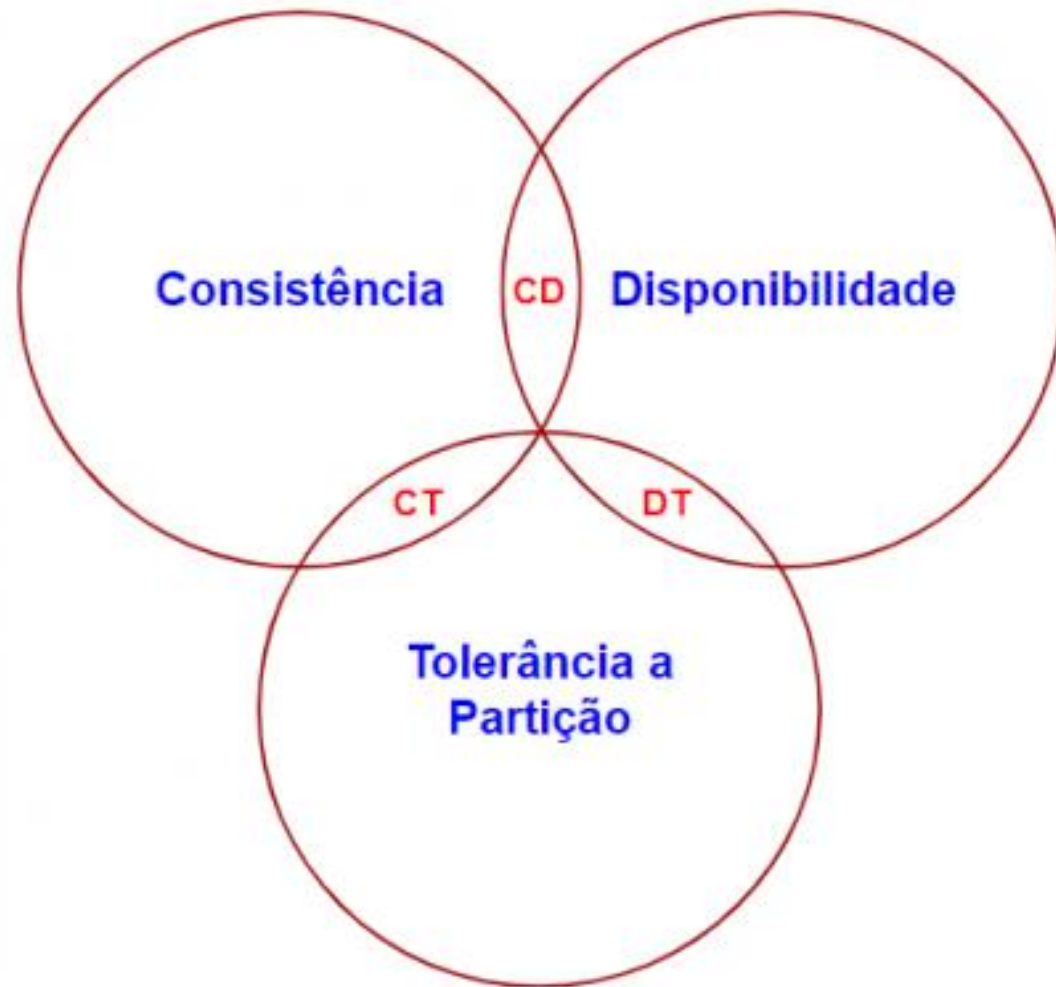
ACID	BASE
Consistência Forte	Consistência Fraca
Isolamento	Foco em Disponibilidade
Transações aninhadas	Repostas Aproximadas
Disponibilidade	Simples / Rápido
Conservador	Agressivo

Teorema CAP

- Uma característica de sistemas de bancos de dados distribuídos é a observação do Teorema CAP:
 - Consistência: Mesma visão dos dados;
 - Disponibilidade: Acesso aos dados sempre disponível;
 - Tolerância à Partição: Manutenção das propriedades independentemente de alterações ou implantações em algum dos nós do cluster.

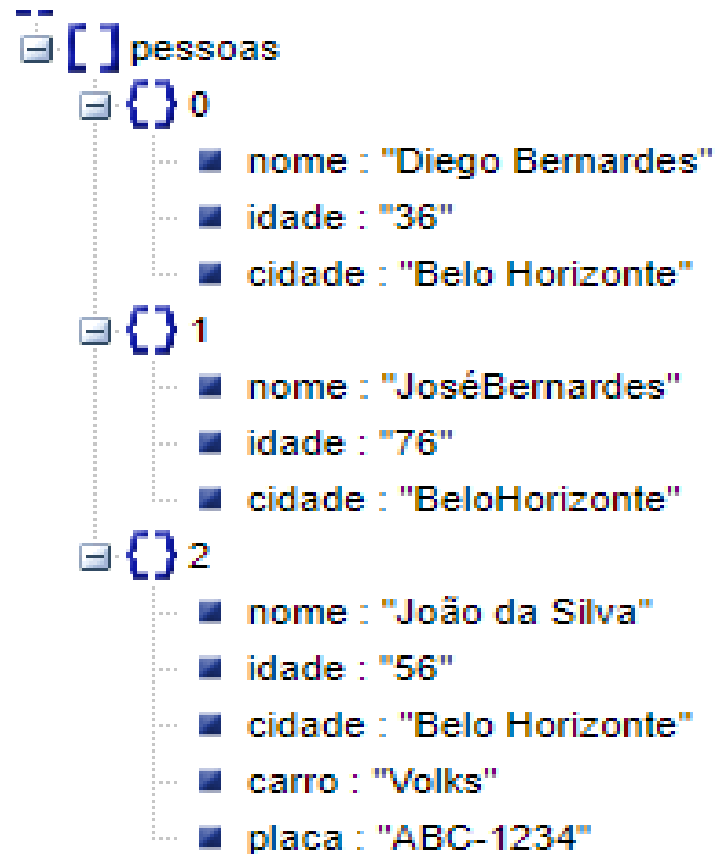
Teorema CAP

iGTi



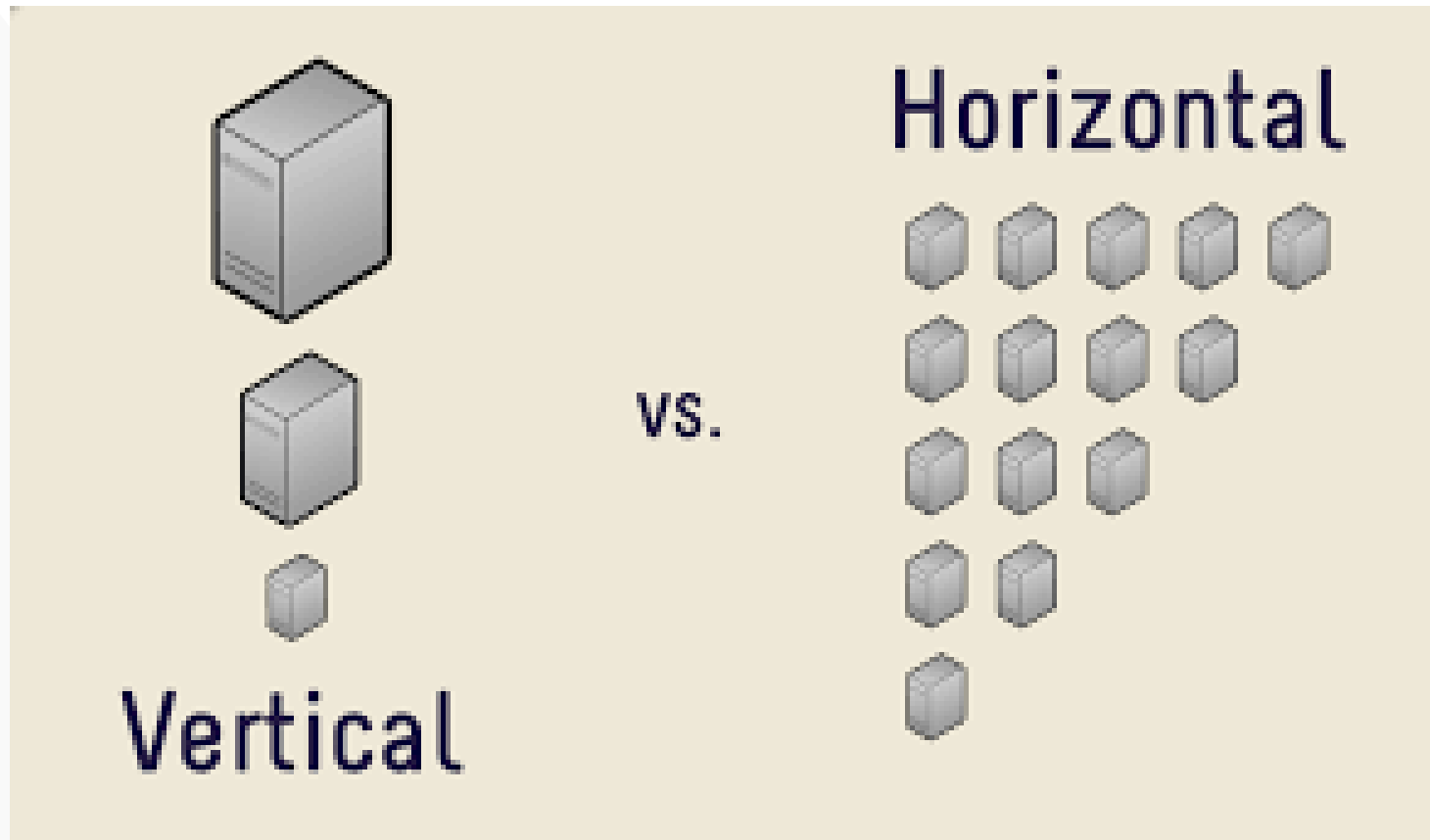
Quando usar NoSQL

- Quando a estrutura de dados é flexível.



Quando usar NoSQL

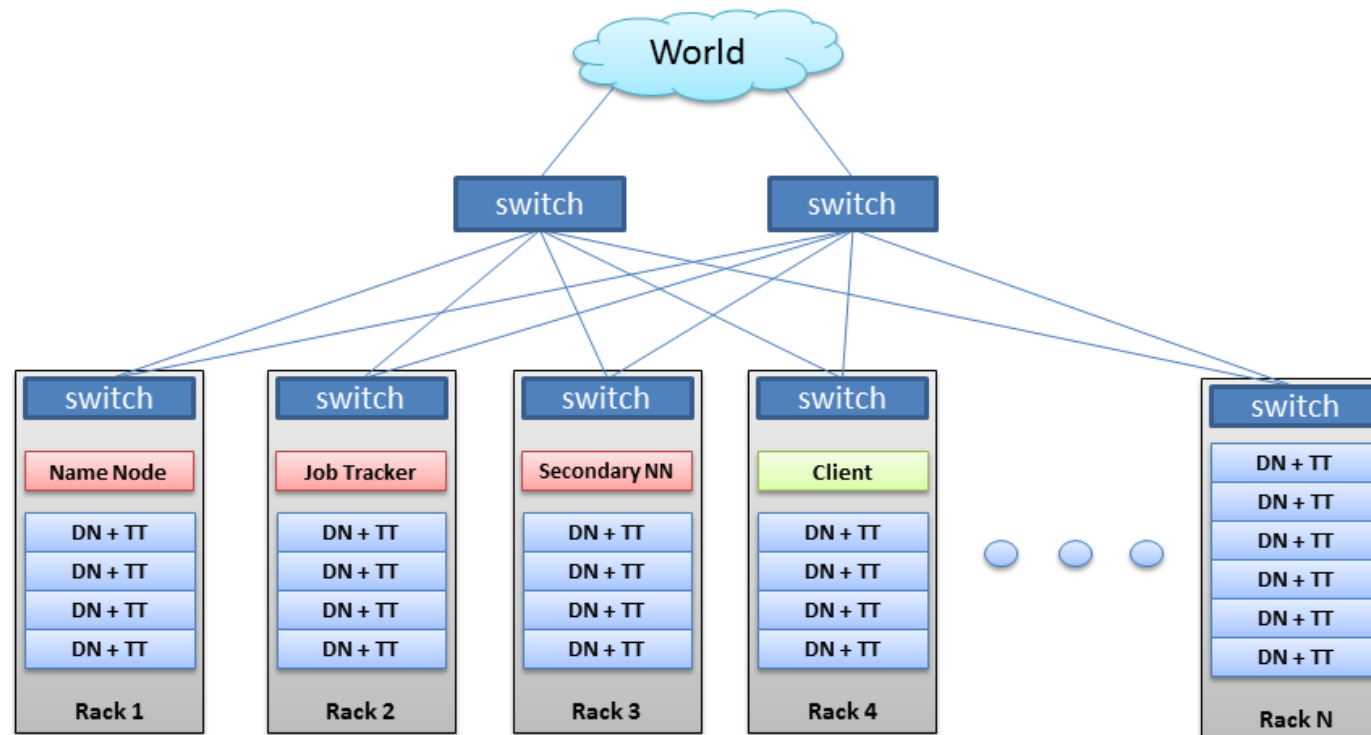
- Quando existe a necessidade de escalabilidade ágil e barata.



Quando usar NoSQL

- Quando disponibilidade é um requisito crítico.

Hadoop Cluster

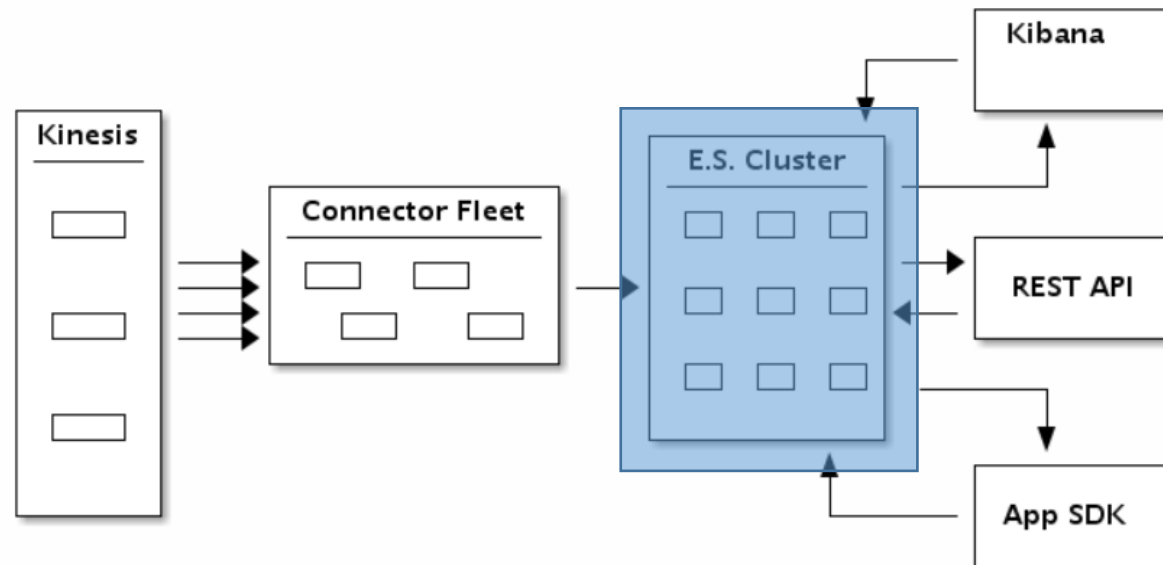


Quando usar NoSQL

- Baixo Custo Operacional:
 - NoSQL geralmente é Free / Open Source;
 - Utilizam clusters com nós de baixo custo;
 - Aumentar o cluster é mais barato do que comprar servidores mais robustos;
 - Não há custo com licença de linguagens de programação ou ambientes integrados.

Quando usar NoSQL

- Funcionalidades Especiais:
 - Índices específicos;
 - Acesso via API Restful.



Tipos de Bancos de Dados NoSQL



- Bancos de Dados Orientados a Colunas:
 - Cassandra, Hbase, Google Big Table.
- Bancos de dados Orientados a Documentos:
 - MongoDB, Couchbase.
- Bancos de dados Chave-Valor:
 - Dynamo, Riak.
- Bancos de dados baseados em Grafos:
 - Neo4J.

Alguns Players que usam NoSQL



- Amazon: Dynamo.
- Apache: CouchDB.
- Linkedin: Voldemort.
- Facebook: Cassandra.
- Twitter: Cassandra.
- Google: Google Big Table.
- New York Times: MongoDB.

Próxima aula...



- ☐ Bancos de Dados Colunares.



Fundamentos de Bancos de Dados

Aula 3.2.1. Introdução aos Bancos de Dados Colunares (Parte 1)

Prof. Diego Bernardes

Nesta aula



- ❑ Introdução aos Bancos de Dados Colunares.

Bancos de Dados Colunares

- **Introdução**

- Um banco de dados colunar é otimizado para recuperação rápida de colunas de dados.
- Geralmente em aplicações analíticas.
- O mecanismo colunar reduz o esforço computacional para operações de entrada/saída de dados em disco.
- O princípio colunar é de trazer apenas os dados que serão utilizados pela aplicação, em detrimento de buscas em linhas em SGBDs tradicionais.



Bancos de Dados Colunares

- **Características**

- Colunas definidas por triplas:
 - Linha, Coluna e Timestamp.
- Supercoluna: Coluna que agrupa outras colunas, formando as “famílias de colunas”.
- Famílias de colunas: Colunas que são armazenadas no mesmo conjunto de arquivos, via de regra dados complementares ou relacionados entre si.
- Bancos colunares não possuem mecanismos de junções.

Bancos de Dados Colunares

- Exemplo:

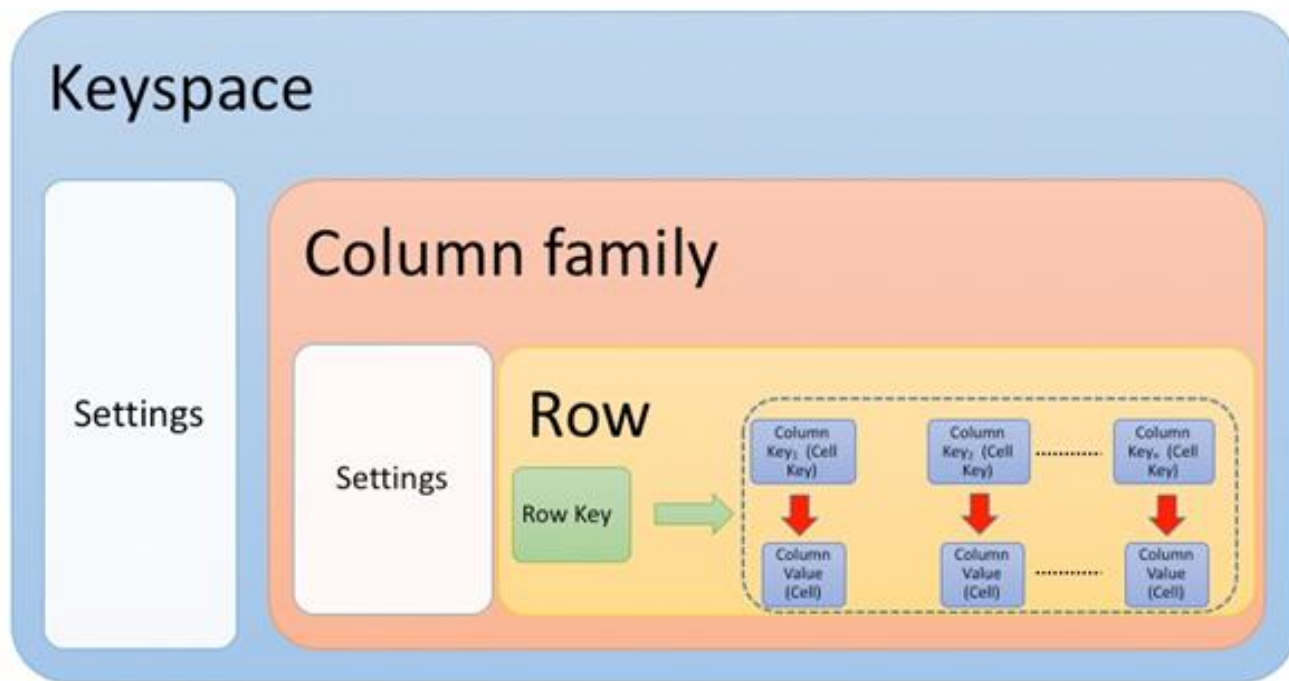
<i>Rowkey</i>	Famílias de Colunas			
	Principais		Adicionais	
	Produto	Tamanho	Tela	Tipo
13487				
13488		20'	LCD	
13489		40'	LED	
13490		50'	PLASMA	SMART
		32'	LED	SMART

Apache Cassandra

- Cassandra é um banco de dados NoSQL / Colunar.
- Informações sobre Cassandra:
 - Escalabilidade e disponibilidade sem pontos de falha.
 - Modelo de dados em famílias de colunas.
 - Bom desempenho em operações de I/O.
 - Linguagem de consulta similar a SQL, facilitando a recuperação de informações.
 - Esquema flexível.
 - Suporte a replicação.

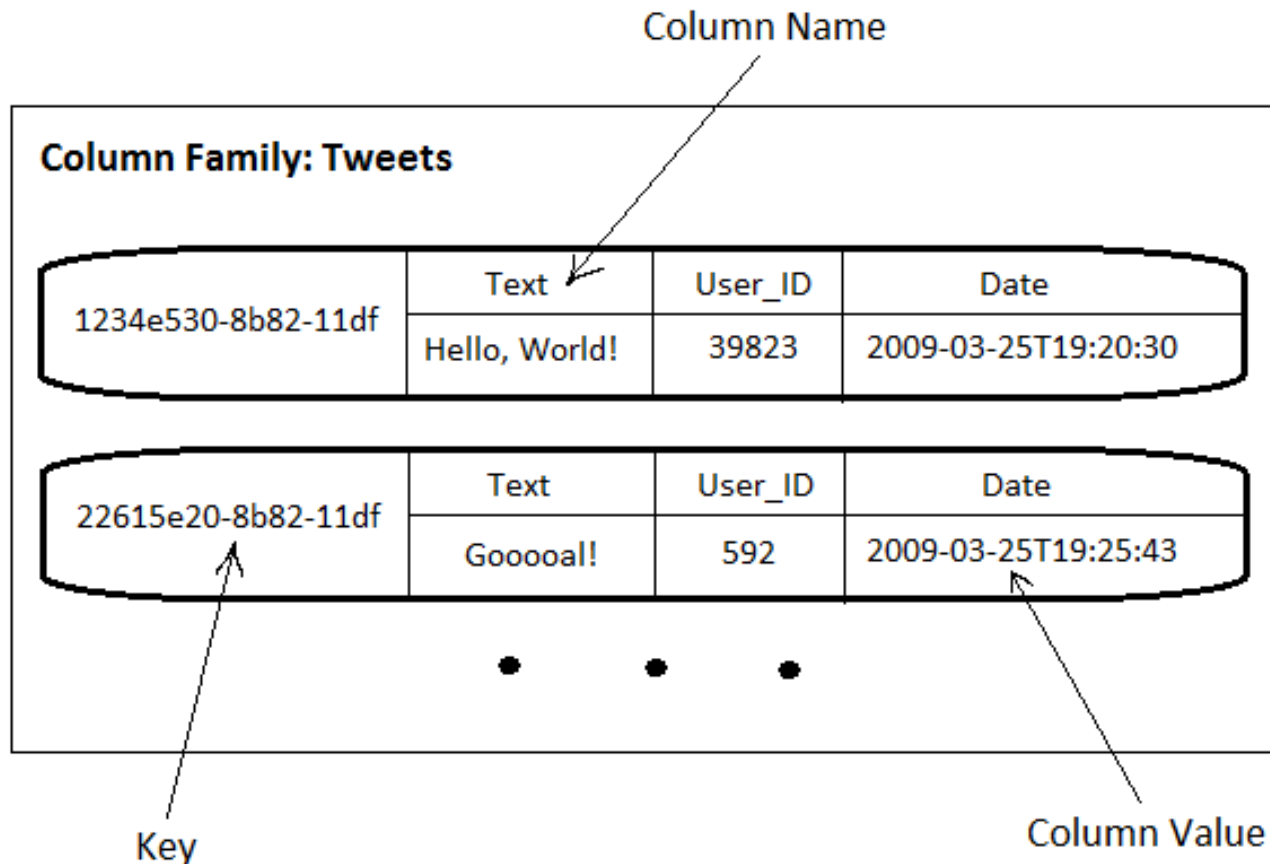
Apache Cassandra

- Arquitetura.



Apache Cassandra

- Modelo de Dados.



Próxima aula...



- ❑ Demonstração do Apache Cassandra.



Fundamentos de Bancos de Dados

Aula 3.2.2. Introdução aos Bancos de Dados Colunares (Parte 2)

Prof. Diego Bernardes

Nesta aula



- ❑ Introdução aos Bancos de Dados Colunares.

Demonstração

IGTi



Próxima aula...



- ☐ Bancos de Dados Colunares.

Fundamentos de Bancos de Dados

Aula 3.3. Introdução aos Bancos de Documentos

Prof. Diego Bernardes

Nesta aula



- ❑ Introdução aos Bancos de Documentos.

Bancos de Dados de Documentos

- **Introdução:**
 - Um banco de dados orientado a documentos é um banco de dados projetado para armazenar, recuperar e gerenciar informações semiestruturadas, denominadas documentos.
 - Cada documento é um arquivo que agrupa relações de chave-valores.
 - Formatos mais comum: JSON.

Bancos de Dados de Documentos

- **Introdução**

- Um banco de dados orientado a documentos é um banco de dados projetado para armazenar, recuperar e gerenciar informações semiestruturadas, denominadas documentos.
- Cada documento é um arquivo que agrupa relações de chave-valores.
- Formatos mais comum: JSON.

Bancos de Dados de Documentos

- **Arquivo JSON**

- JSON (JavaScript Object Notation) é um modelo para armazenamento e transmissão de informações no formato texto.
- Formato de arquivo inicialmente criado para armazenamento compacto de texto para interoperabilidade entre sistemas.
- Sintaxe:
 - “curso”: “banco de dados”
 - “idade”: 35
 - “disciplinas”: [“Introdução”, “Consultas SQL”, “DDL”]

Bancos de Dados de Documentos

- Exemplo:

```
{
  "nota_fiscal": [
    {
      "id_compra": 123,
      "valor": 150.00,
      "produto": "Televisão"
    },
    {
      "id_compra": 123,
      "valor": 150.00,
      "produto": "Mesa de Jantar",
      "itens": [
        "cadeira 01",
        "cadeira 02",
        "cadeira 03",
        "cadeira 04",
        "tampo de vidro"
      ]
    }
  ]
}
```

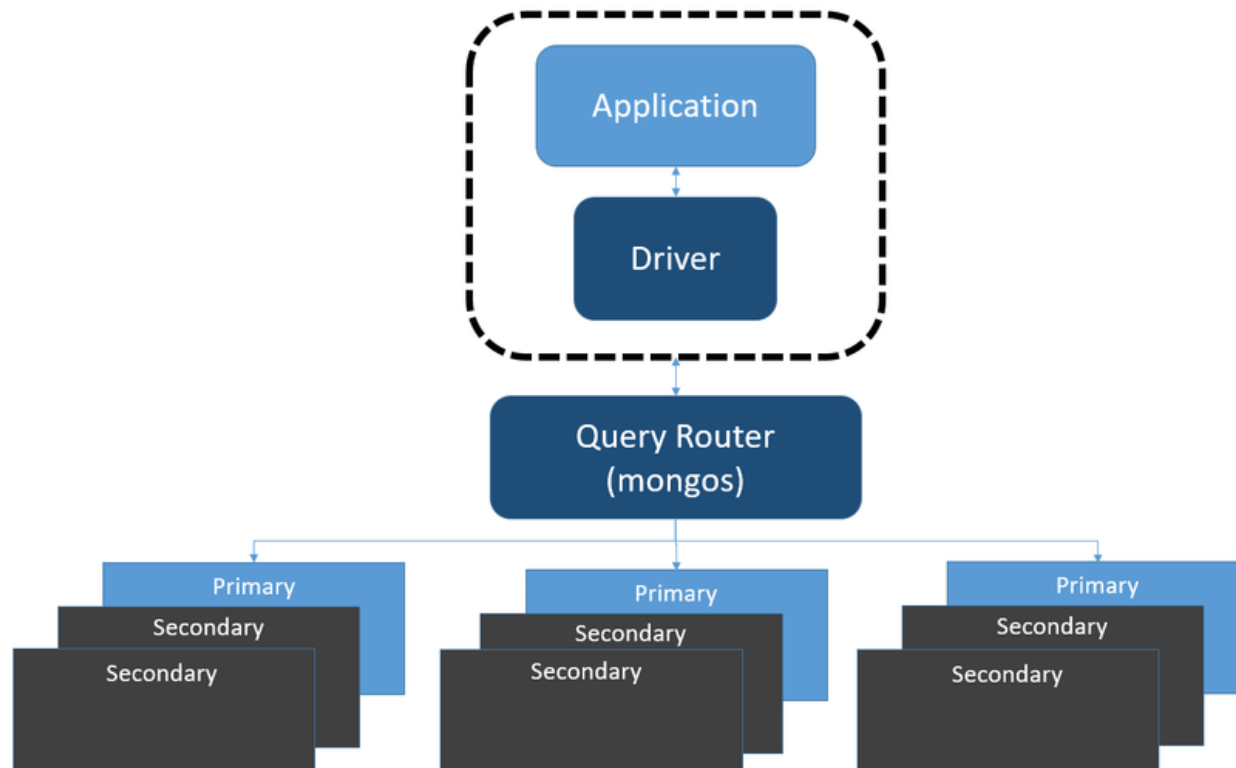
Bancos de Dados de Documentos

- **Aplicações**

- Mecanismo de armazenamento interessante quando a informação está contida em um único local: documento.
- Suporte a aplicações com uso de API Rest.
- Suporte a escalabilidade horizontal.
- Buscas utilizando processamento paralelo.

Bancos de Dados de Documentos

- Arquitetura.



Fonte: <https://www.researchgate.net/>

Bancos de Dados de Documentos



Case Study: The New York Times Runs MongoDB

Perhaps your business has settled on the exact right operating model, one that will remain static for years, if not decades. But for the 99.999 percent of the rest of the world's enterprises, your market is in a constant state of flux, demanding constant iterations on how you do business. As the Research & Development group of [The New York Times Company](#) (NYT) has [found](#), a key way to confront the constant flux of today's businesses is to build upon a flexible data infrastructure like MongoDB.

The story behind the The New York Times Company's use of MongoDB isn't new. Data scientist and then NYT employee [Jake Porway](#) spoke in [June 2011](#) about how the media giant uses MongoDB in Project Cascade, a visualization tool that uses MongoDB to store and manage data about social sharing activity related to NYT content.

Fonte: <https://www.mongodb.com/>

Bancos de Dados de Documentos

Exemplo Hipotético da Aplicação – The New York Times

```
{
  "interactionId": "f35ace79b903eedfe5198f386d6fda0c",
  "subscriptionId": "ABC123456a891e3406789123216789",
  "hash": null,
  "hashType": null,
  "interaction": {
    "schema": {
      "version": 3
    },
    "source": "Twitter for Android",
    "type": "twitter",
    "created_at": "Thu, 09 May 2013 08:59:46 +0000",
    "content": "RT @Real_Liam_Payne: Thank you denmarkkkk :) love youuuu :)",
    "id": "f35ace79b903eedf75198f386d6fd40b",
    "author": {
      "hash_id": "c6add0a675830a785598a513d586203c"
    },
    "tags": [
      "type.share",
      "demo.tag",
      "another.one",
      "foobar"
    ]
  }
}
```

Próxima aula...



- ❑ Demonstração do MongoDB.