Faculdade XPe

Fundamentos de Bancos de Dados

Capítulo 1.
Introdução

Prof. Diego Bernardes

Faculdade XPe

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

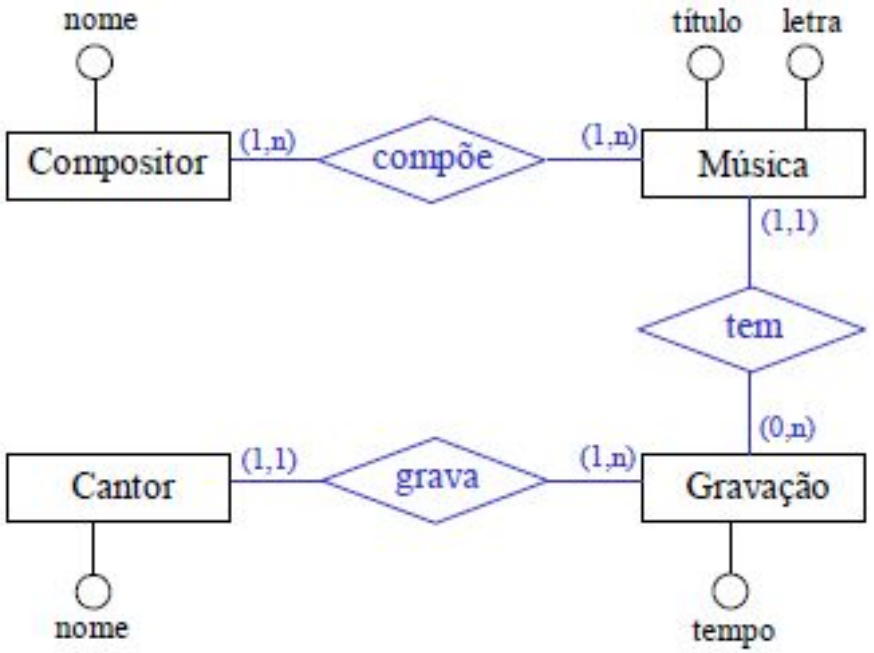
  O nível lógico é o nível intermediário, neste nível é possível representar todo o banco de dados, com suas estruturas e relacionamentos, mas sem a preocupação de como o dado será gravado no disco.

- 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.

Faculdade XPe

Fundamentos de Bancos de Dados

Aula 1.2. Introdução à Modelagem de Dados

Prof. Diego Bernardes

Nesta aula

- Introdução à Modelagem de Dados.

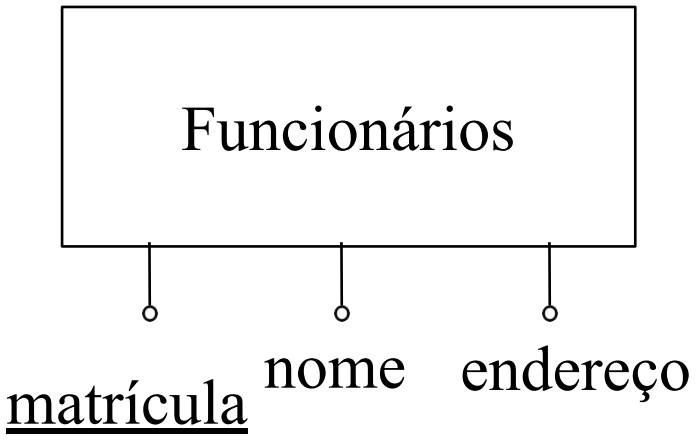
Modelagem Entidade-Relacionamento

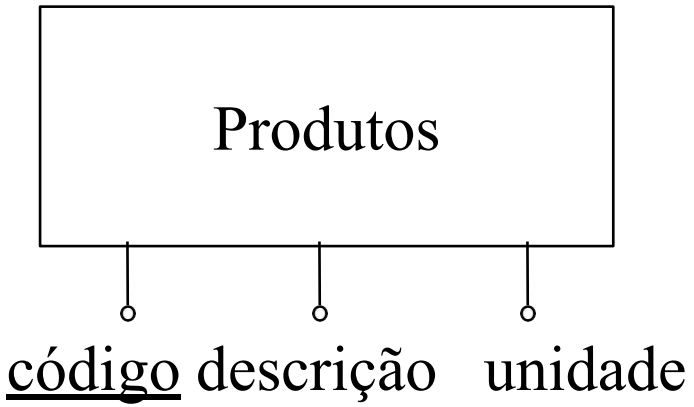
- Elementos:
  - Entidades
    - Conjuntos de “elementos” que possuem características próprias.
  - Atributos
    - Representam as características de uma Entidade.
  - Relacionamentos
    - Vínculos ou associações entre Entidades.

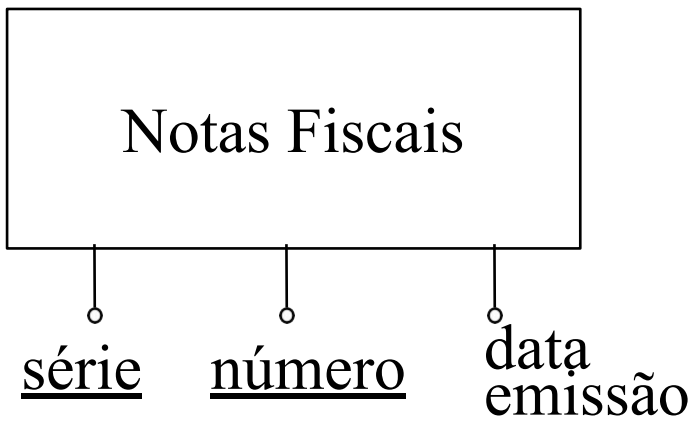
Entidade

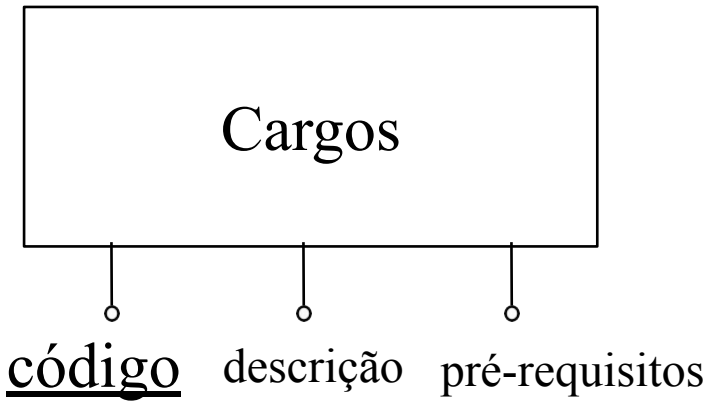
- Conjunto de objetos sobre os quais é preciso armazenar informações.
- Conjunto de vários elementos (mais que 1).
- Conjuntos de elementos distinguíveis que aceitam um código para diferenciá-los.
- Seus atributos NÃO dependem de outras entidades.
- Exemplos: de possíveis Entidades:
  pessoas, locais,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

- Entidade


Funcionários
matrícula
nome
endereço


Produtos
código
descrição
unidade


Notas Fiscais
série
número
data emissão


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.

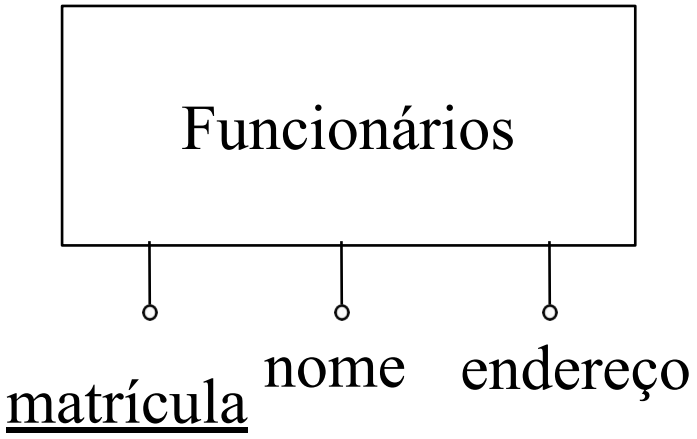
Atributos

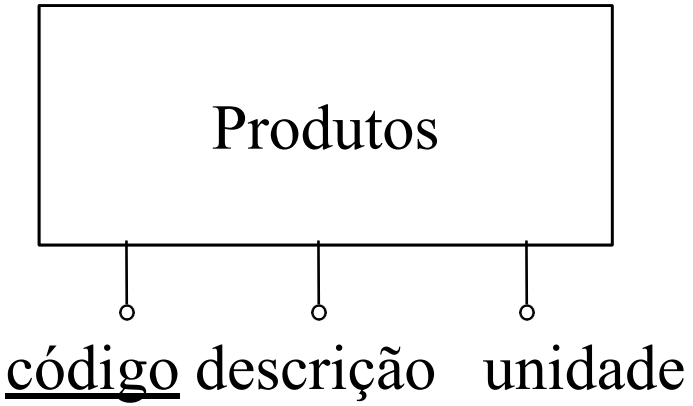
- Chave:
  - seu valor representa um elemento da entidade.
  - seu valor é único para a entidade.
  - deve ser sublinhado.

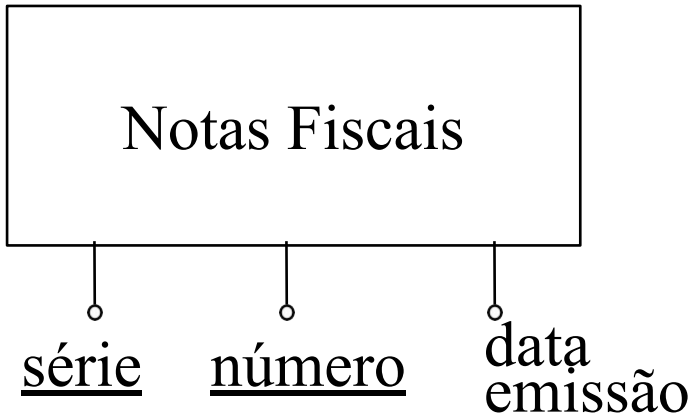
- Composto:
  - necessita ser dividido em sub-atributos, para que seu significado seja melhor compreendido.

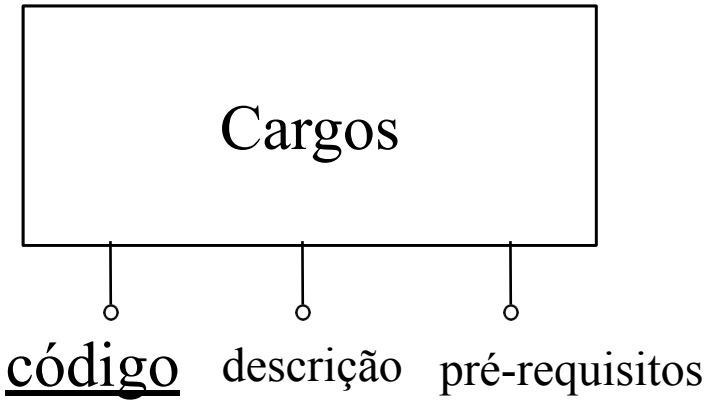
- Multi-valorado:
  - Pode assumir mais do que um valor para cada entidade.

Atributos

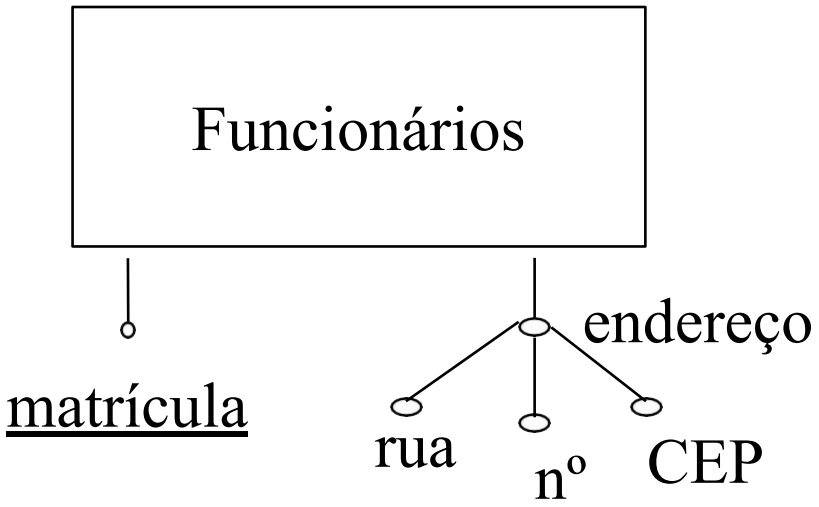

Funcionários
matrícula
nome
endereço


Produtos
código
descrição
unidade

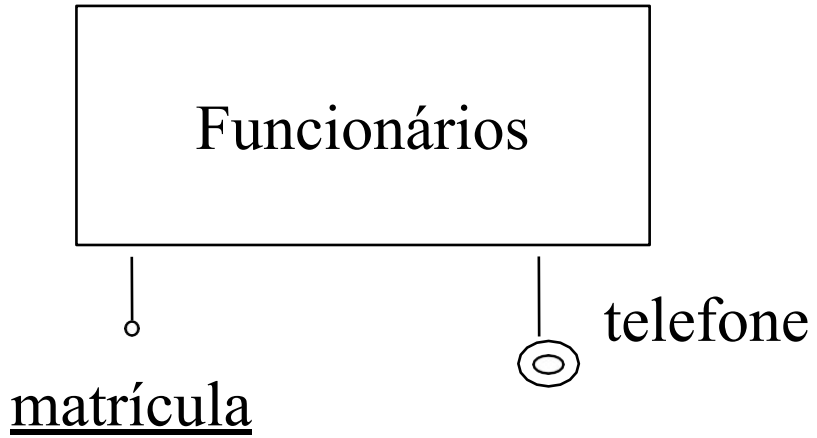

Notas Fiscais
série
número
data emissão


Cargos
código
descrição
pré-requisitos

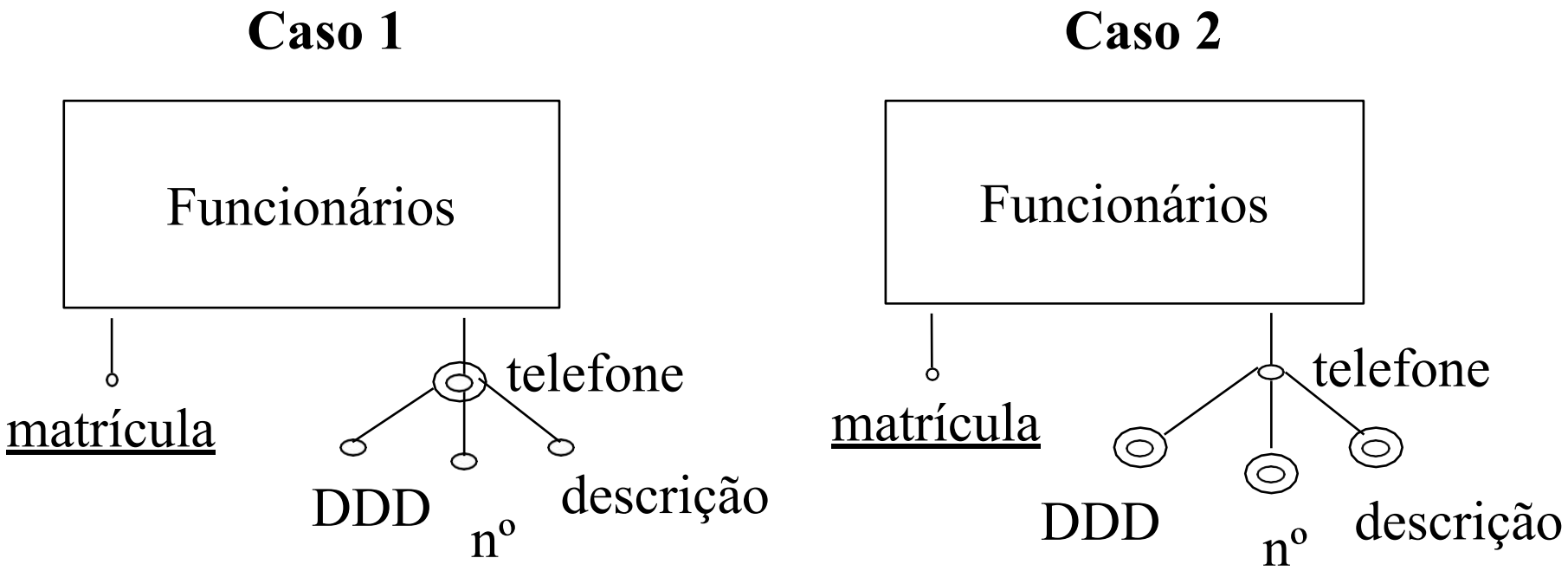
Atributos Compostos



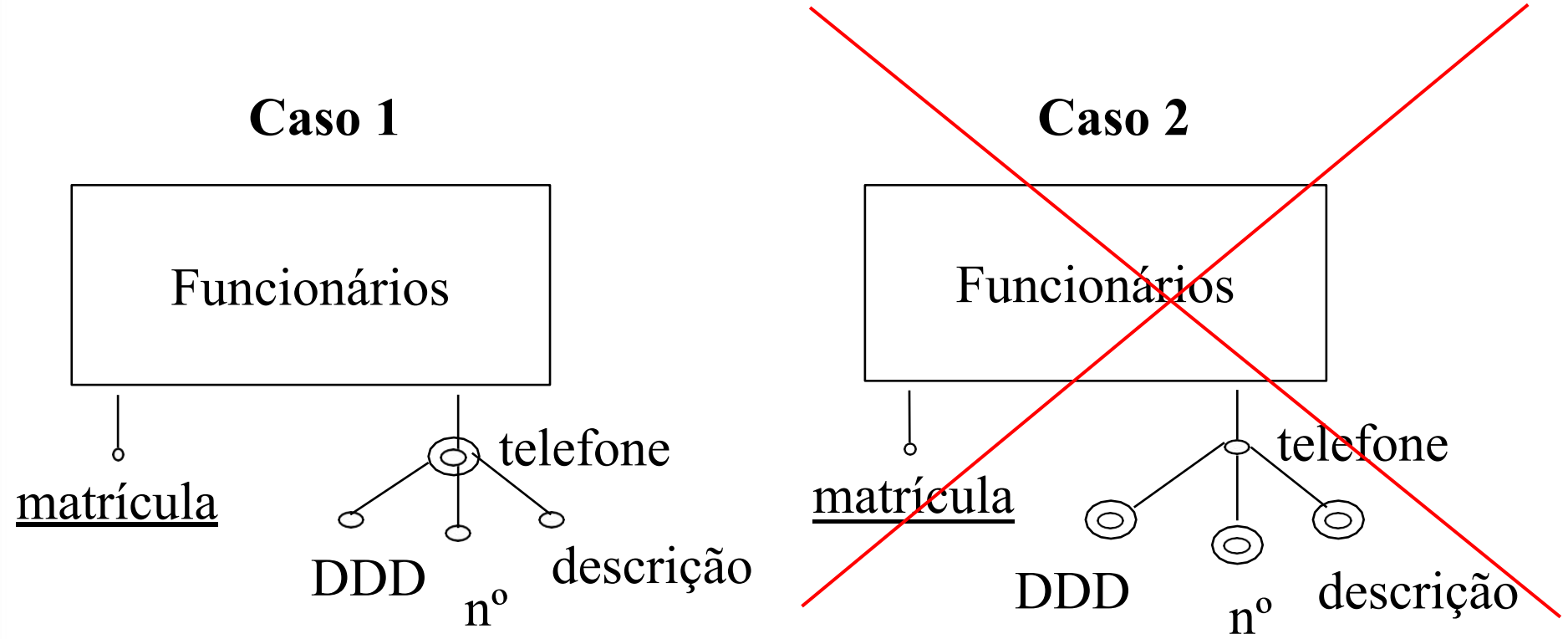
Atributos Multivalorados



Qual solução adotar?

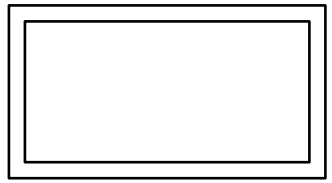


Qual solução adotar?



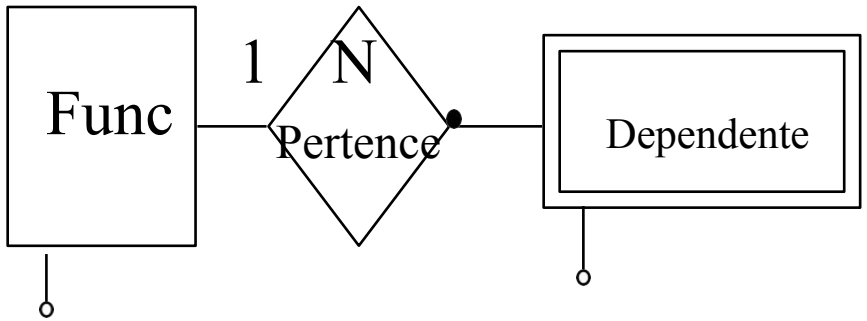
Entidades Fracas

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

Representação: 

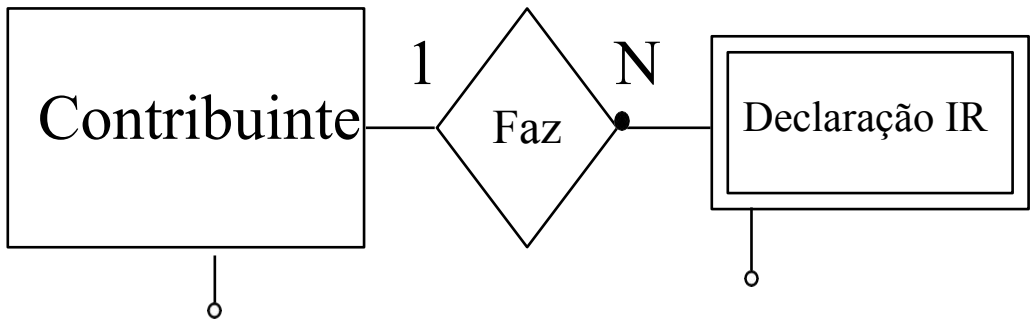
Entidades Fracas

- Necessária Dependência de Existência.



Entidades Fracas

- Necessária Dependência de Identificador.



Próxima aula...

- Introdução à Modelagem de Dados – Relacionamentos entre entidades.

Faculdade XPe

Fundamentos de Bancos de Dados

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

Prof. Diego Bernardes

Nesta aula

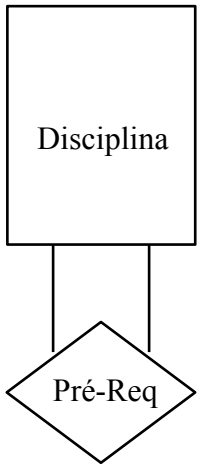
- Modelagem de Dados – Relacionamentos.

Modelagem ER - Relacionamentos

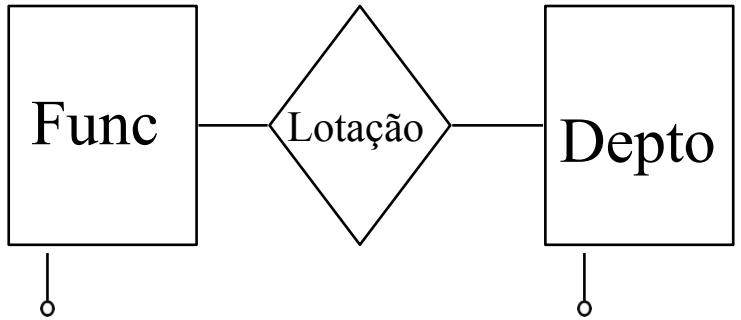
- Associação entre entidades.
- Representam os vínculos que existem entre as entidades no mundo real.
- São representados por losangos.
- Ex.: Em um sistema de controle acadêmico o relacionamento MATRÍCULA, vincula um ALUNO a uma DISCIPLINA.

Modelagem ER - Relacionamentos

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



Grau 1



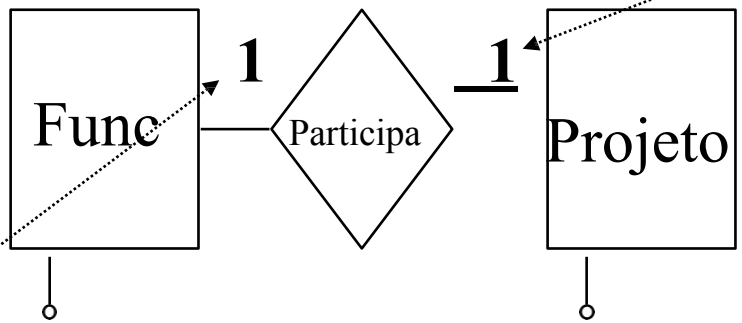
Grau 2

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

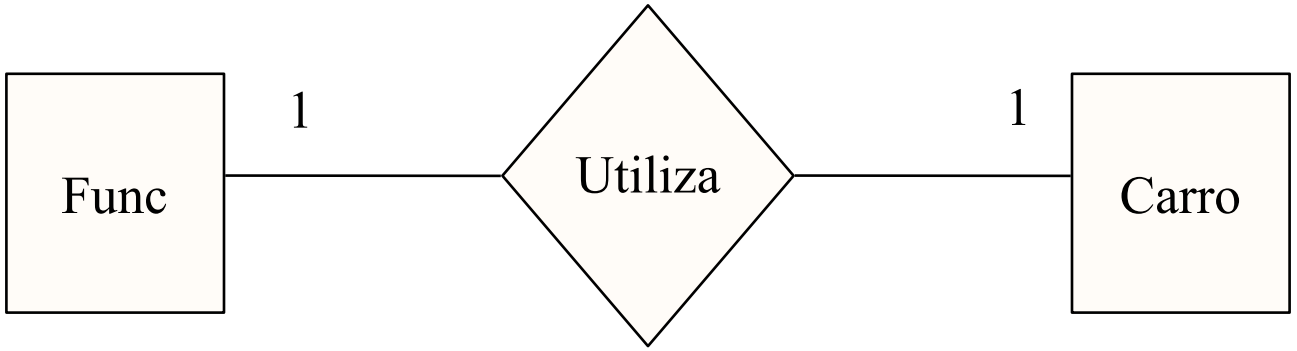
Modelagem ER - Relacionamentos

Cada FUNC participa de quantos PROJETOS?



Cada PROJETO tem a participação de quantos FUNC?

Classe 1:1



Kurt ——— Fargo

Brian ——— Mustang

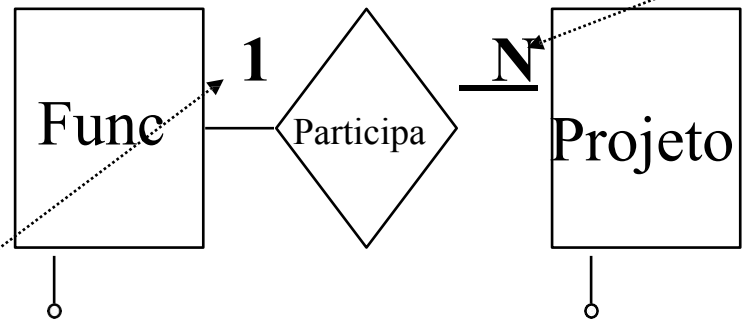
Tonya ——— Ranger

Scott ——— Jeep

Nancy ——— Prizm

Modelagem ER - Relacionamentos

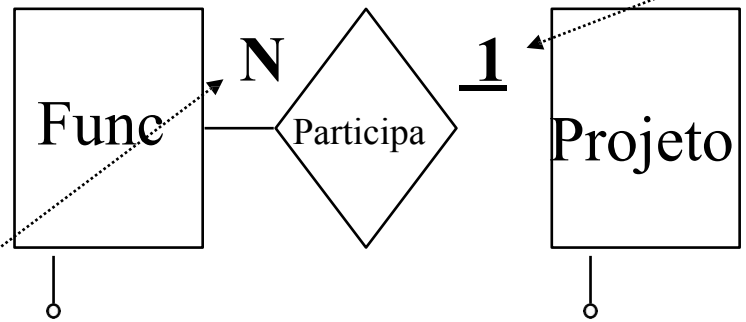
Cada FUNC participa de quantos PROJETOS?



Cada PROJETO tem a participação de quantos FUNC?

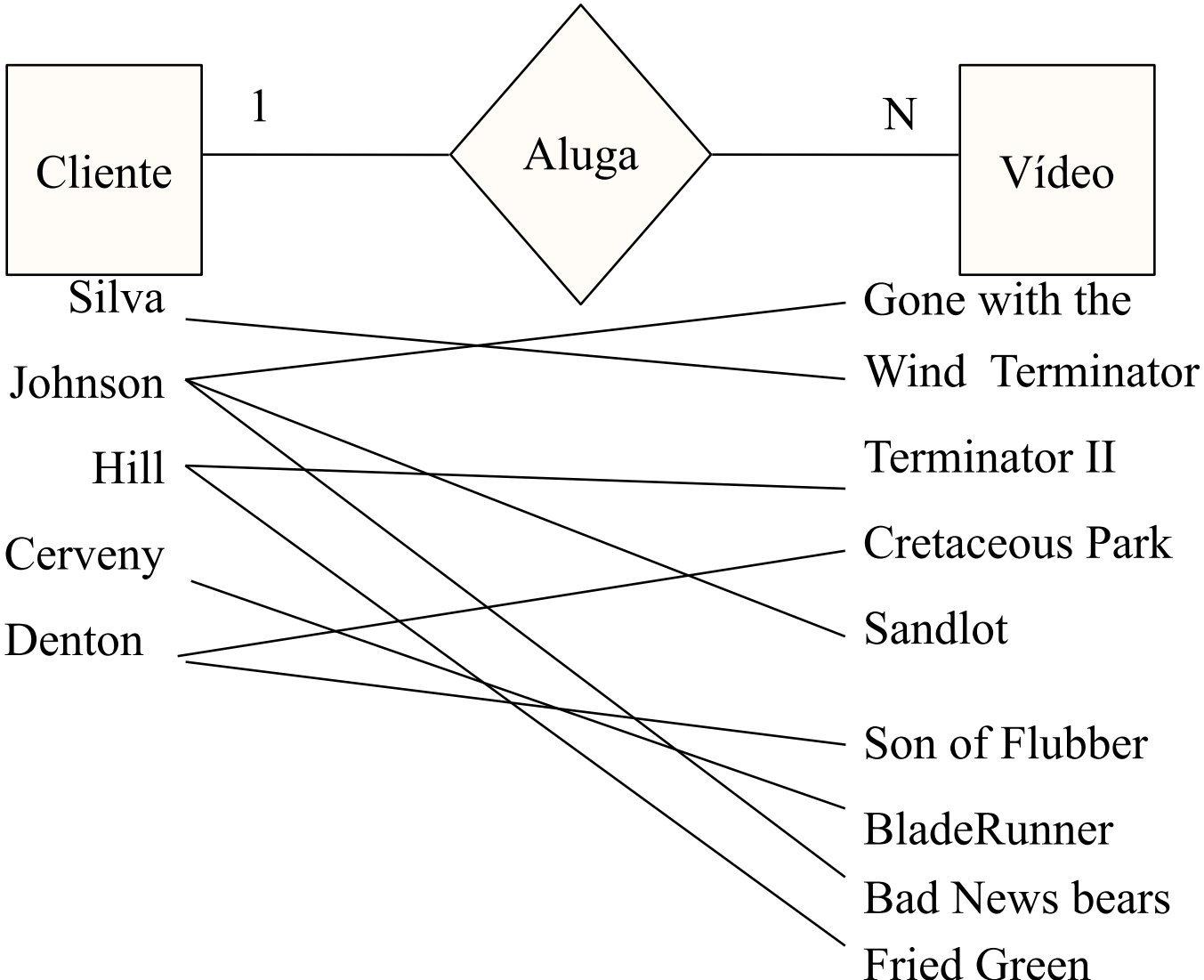
Modelagem ER - Relacionamentos

Cada FUNC participa de quantos PROJETOS?



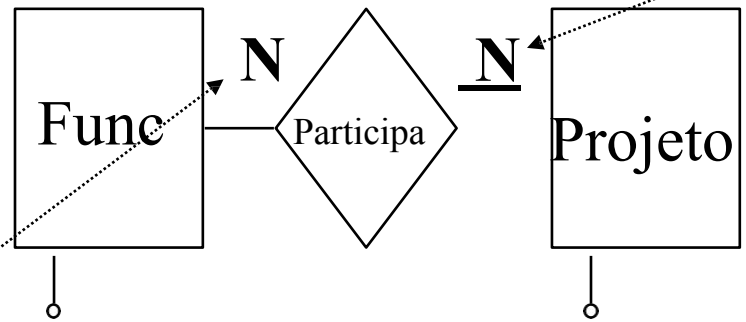
Cada PROJETO tem a participação de quantos FUNC?

Classe 1:N


Cliente
1
Aluga
N
Vídeo
Silva
Johnson
Hill
Cerveny
Denton
Gone with the Wind
Terminator
Terminator II
Cretaceous Park
Sandlot
Son of Flubber
BladeRunner
Bad News bears
Fried Green

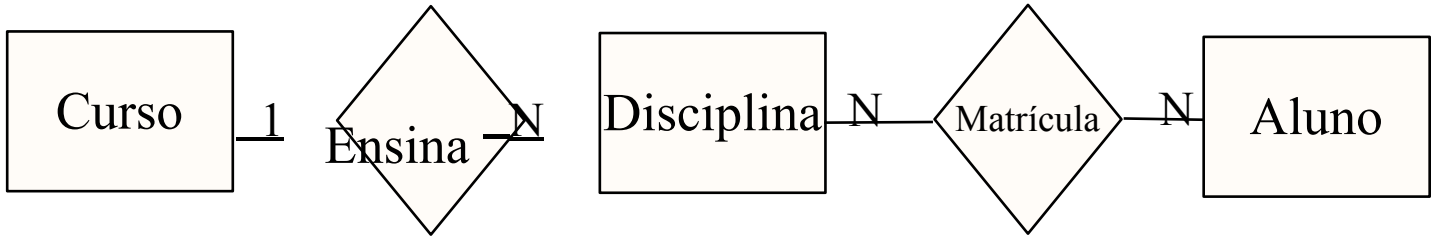
Modelagem ER - Relacionamentos

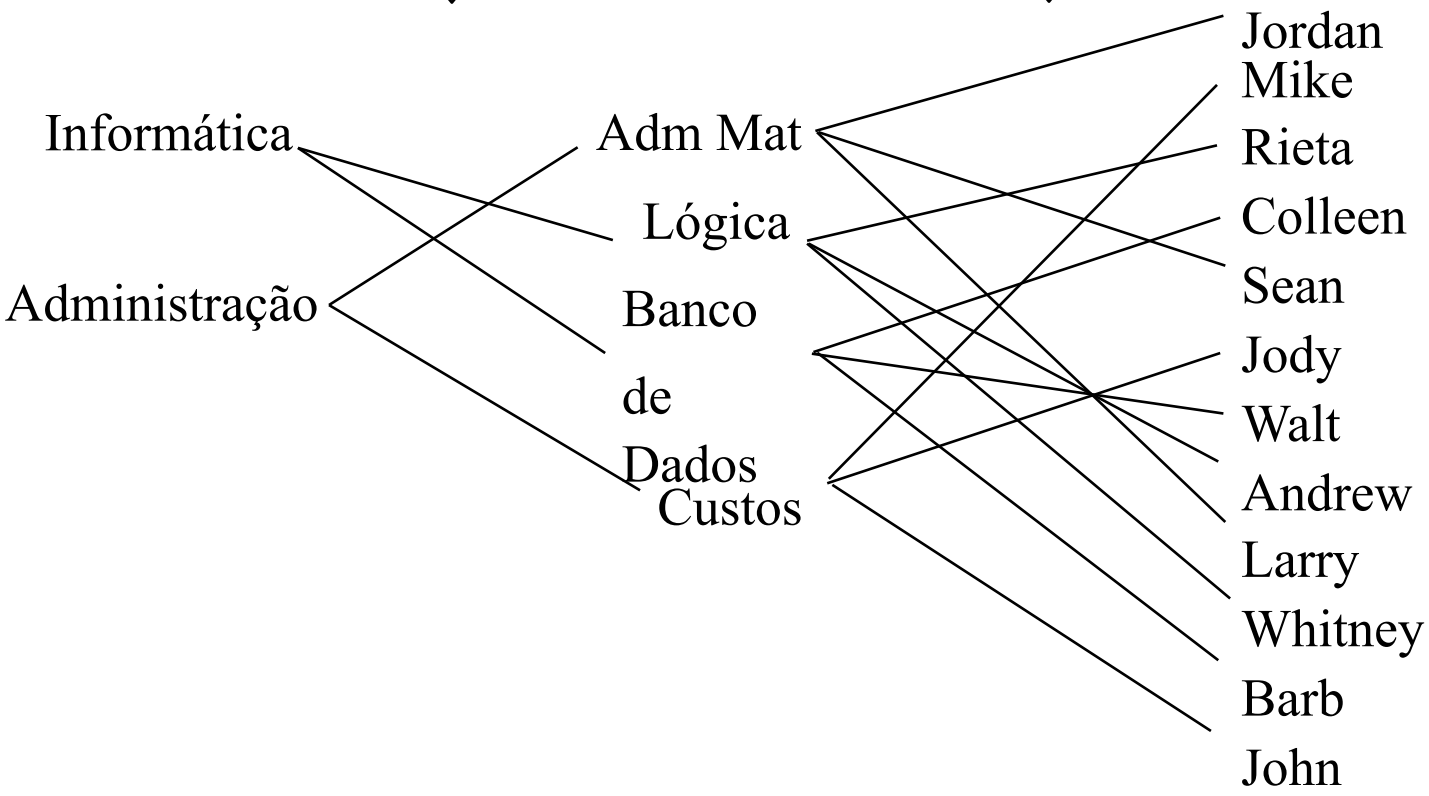
Cada FUNC participa de quantos PROJETOS?



Cada PROJETO tem a participação de quantos FUNC?

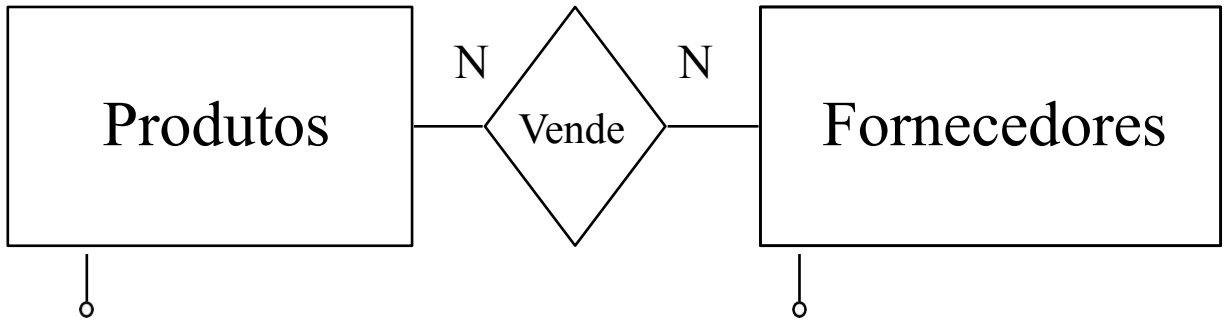
Classes 1:N e N:N





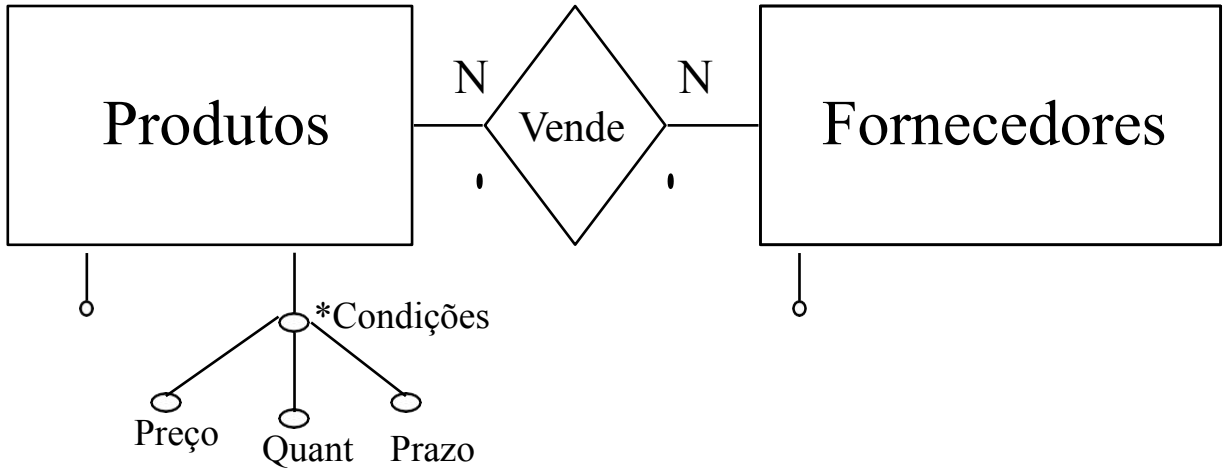
Atributos de Relacionamentos

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



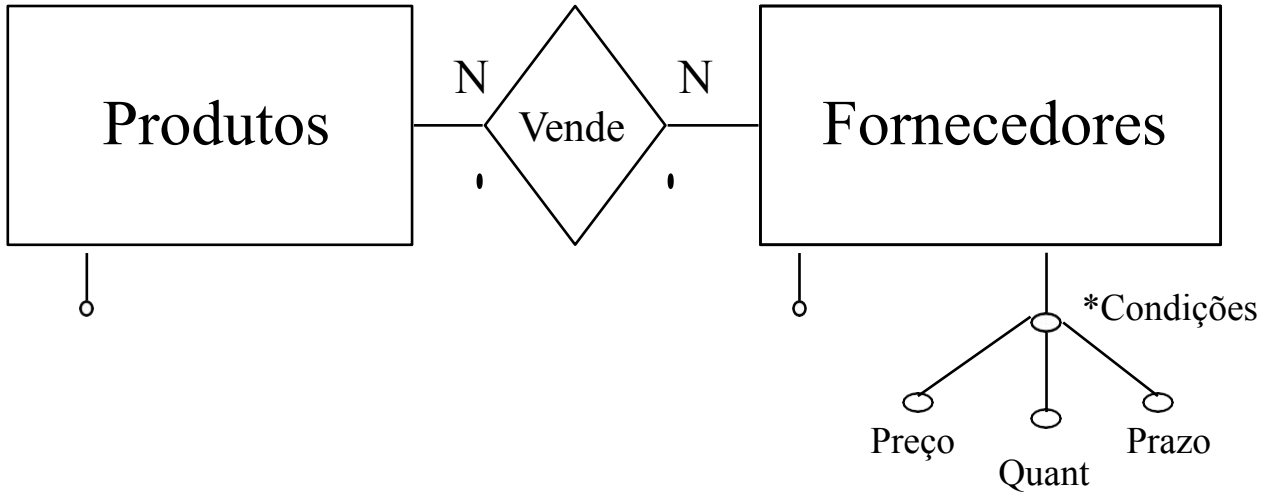
Atributos de Relacionamentos

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



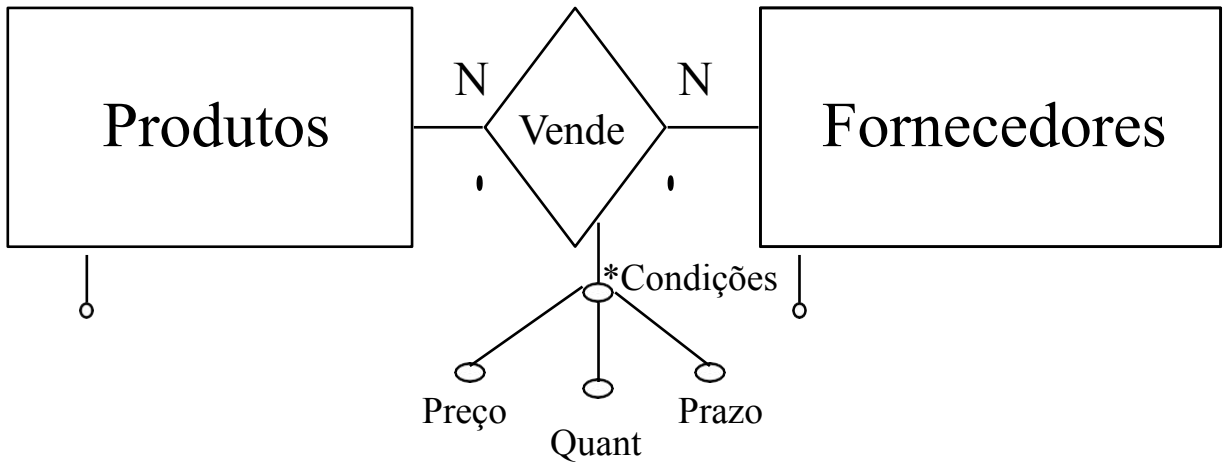
Atributos de Relacionamentos

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



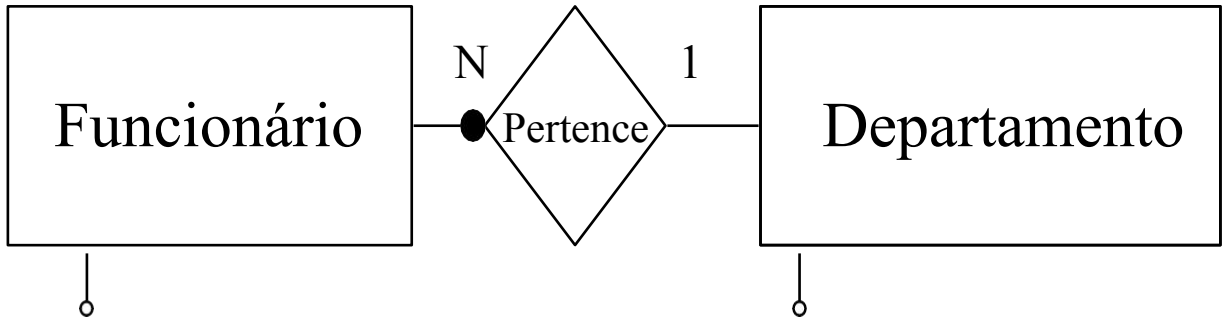
Atributos de Relacionamentos

- 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.

Faculdade XPe

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.

Faculdade XPe

Fundamentos de Bancos de Dados

Capítulo 2. Modelagem de Dados Relacionais

Prof. Diego Bernardes

Faculdade XPe

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	CodDepto	CategFuncional
E5	Souza	D1	C5
E3	Santos	D2	C5
E2	Silva	D1	C2
E1	Soares	D1	C6

Li *nha ou*
Tupla ou
Re *gistro*

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

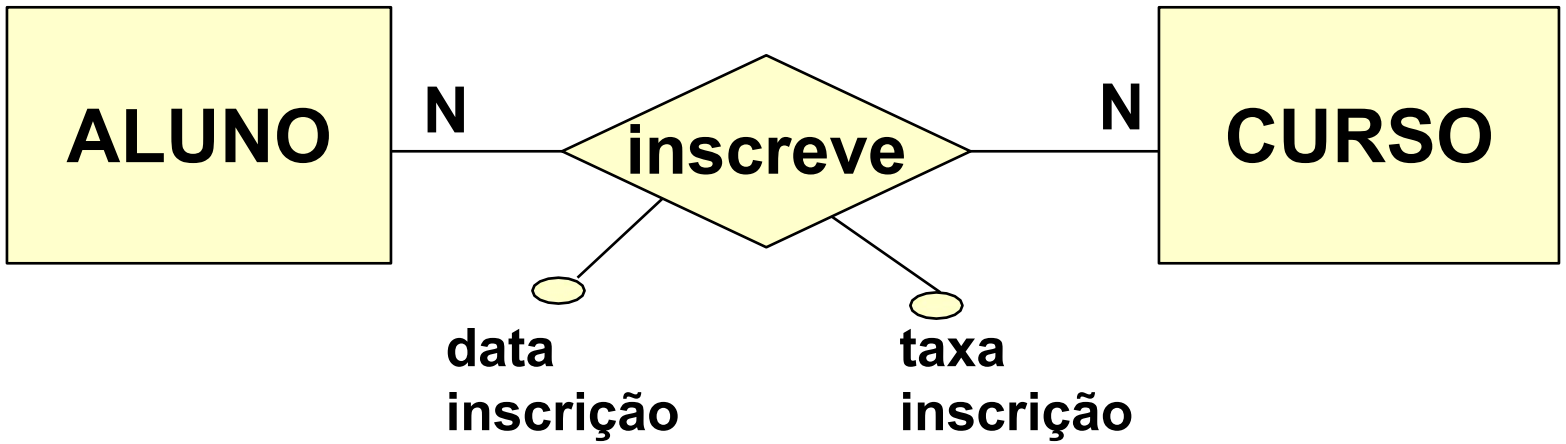
- A tradução do relacionamento depende da cardinalidade das entidades que participam do mesmo.

- Formas básicas de tradução:
    - Tabela própria para o relacionamento.
    - Colunas adicionais dentro da tabela de entidade.
    - Fusão das entidades em uma.

Relacionamentos NxN

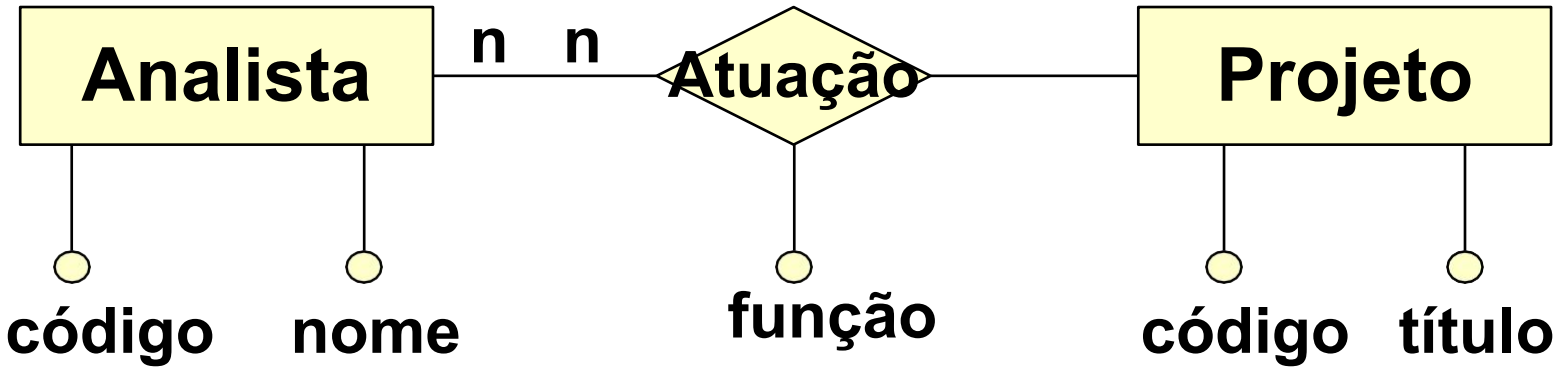
- Para o relacionamento entre A e B de N:N, temos a regra:
  - Criar sempre uma tabela C, agregando as chaves estrangeiras de A e B para formar a chave primária da tabela C, referente ao relacionamento.
  - Caso existam atributos, alocá-los à tabela C como campos normais (descritivos).

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);

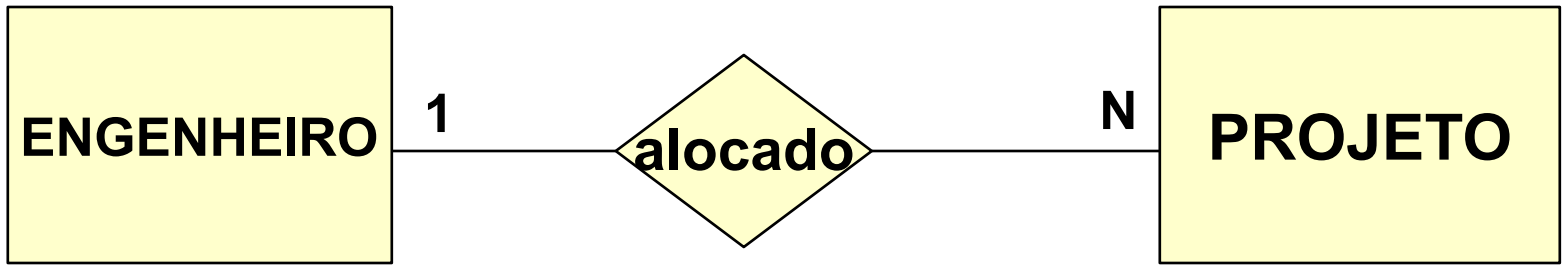
Projeto(codproj, título);

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



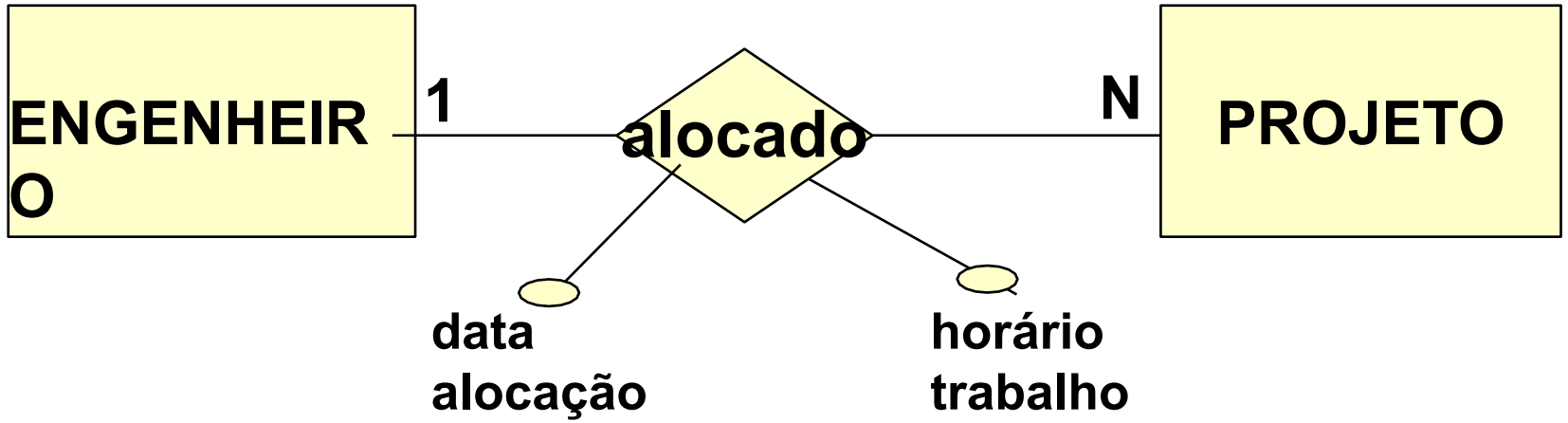
Solução:
Migrar a chave primária da tabela ENGENHEIRO para a tabela PROJETO

(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.

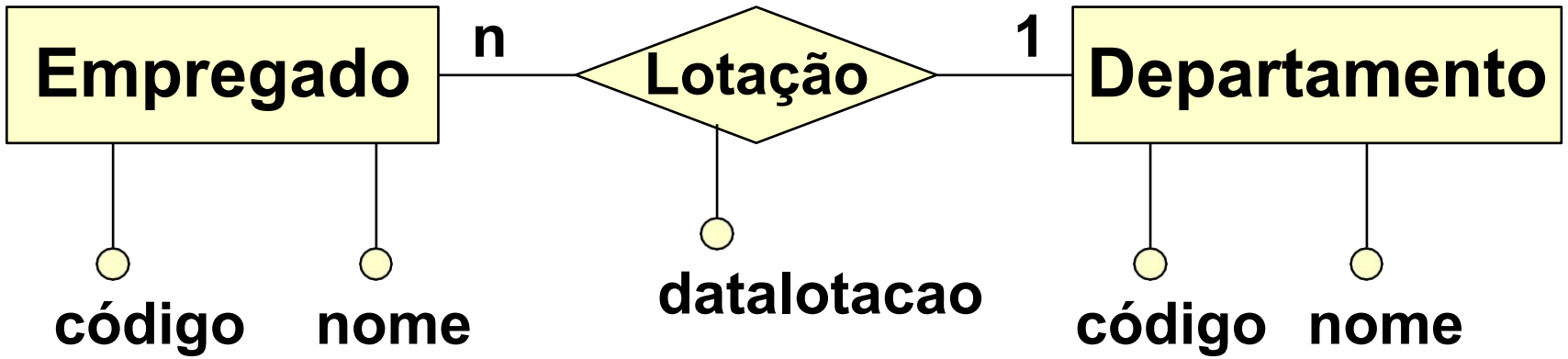
Relacionamentos 1xN



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:

codemp	nome	coddepto	datalotacao
101	João	1	30/12/1976
102	José	2	12/06/2001
103	Maria	1	21/03/1987

Departamento:

coddepto	nome
1	Gerência
2	Vendas
3	Compras

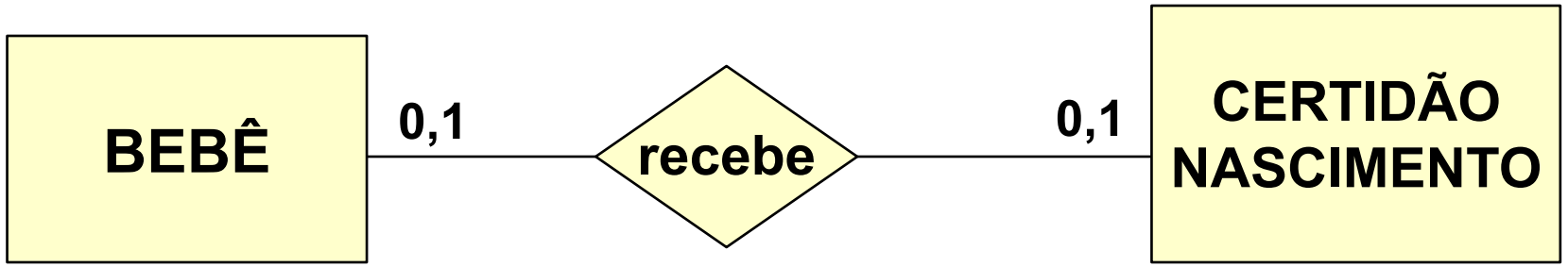
Relacionamentos 1x1

- 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

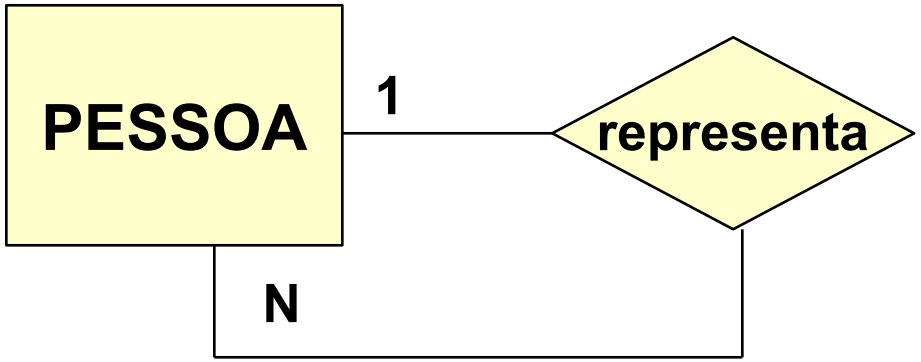
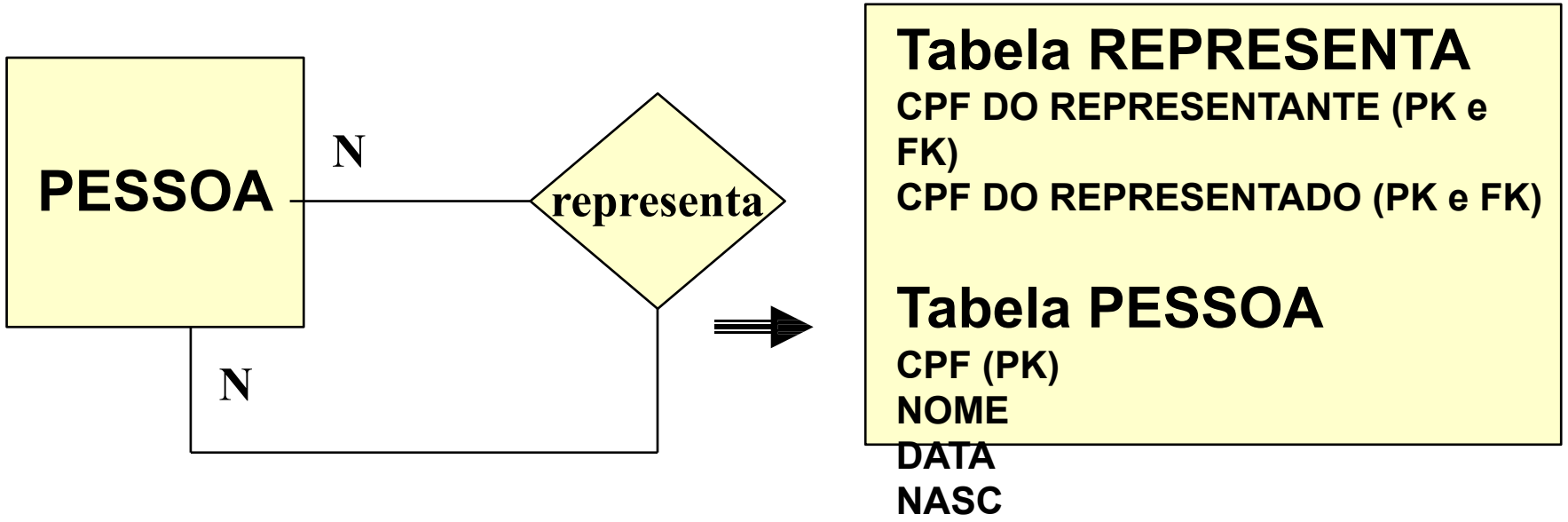


Tabela PESSOA
CPF (PK)
NOME
DATA
NASC
CPF DO REPRESENTANTE (FK)

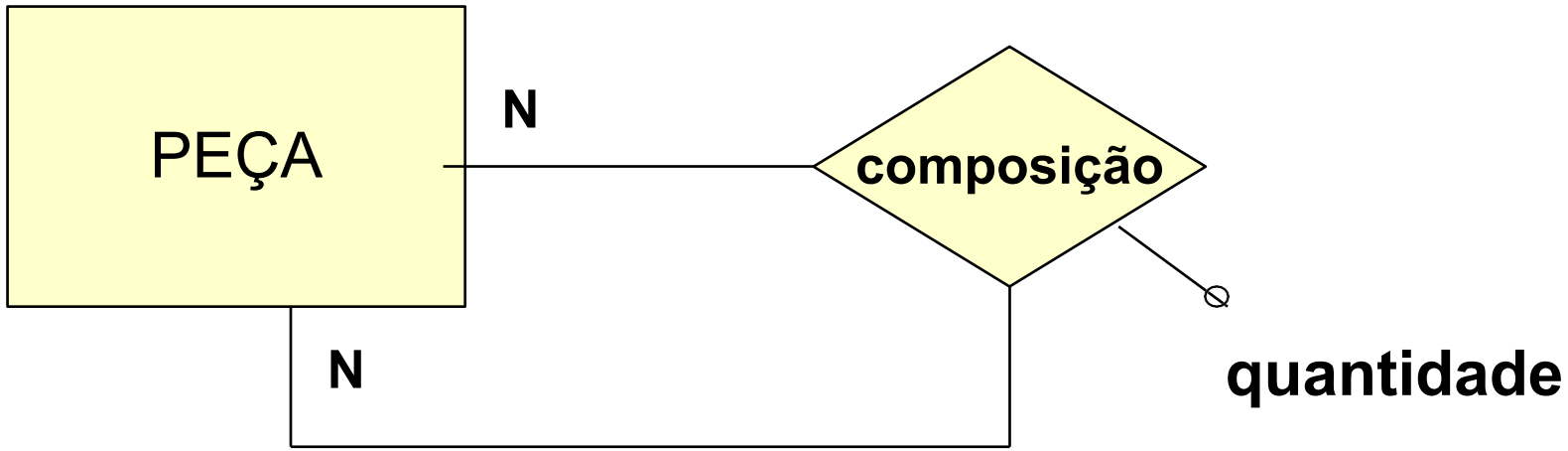
Solução: Migrar a chave primária de PESSOA para PESSOA (como chave estrangeira).

Auto Relacionamento



Solução:
Criara tabela REPRESENTAcom 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_compoe,

quantidade).

Próxima aula...

- Modelo Lógico de Dados.

Faculdade XPe

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.

Faculdade XPe

Fundamentos de Bancos de Dados

Capítulo 3. Bancos de Dados Não Relacionais

Prof. Diego Bernardes

Faculdade XPe

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

- **N**ot **O**nly **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?

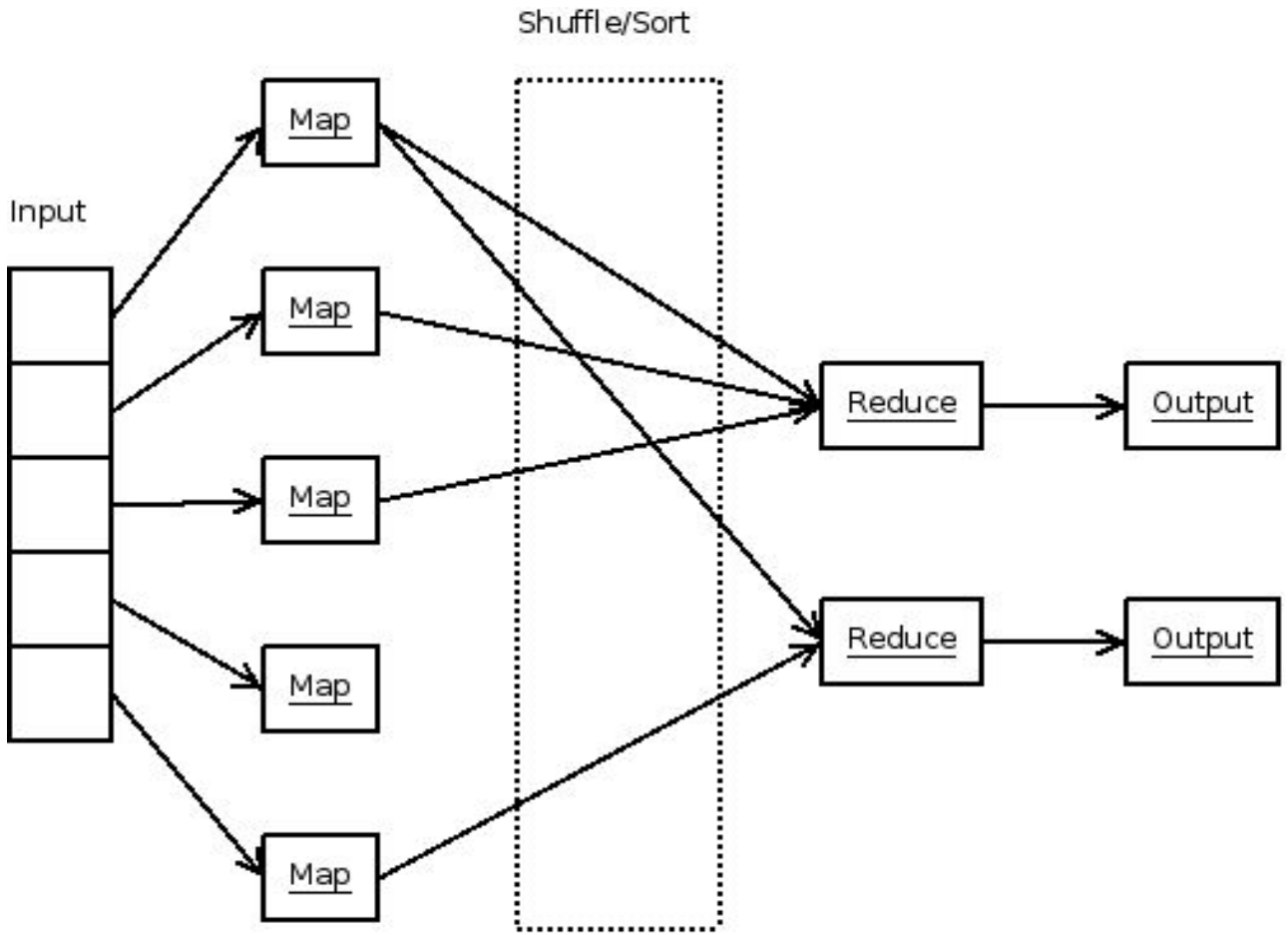
- Dificuldade de escalabilidade dos Bancos de Dados Relacionais.
  - Bancos de dados relacionais escalam, mas quanto maior o tamanho, mais custoso se torna essa escalabilidade, seja pelo custo de novas máquinas, seja pelo aumento de especialistas nos bancos de dados utilizados.

Características

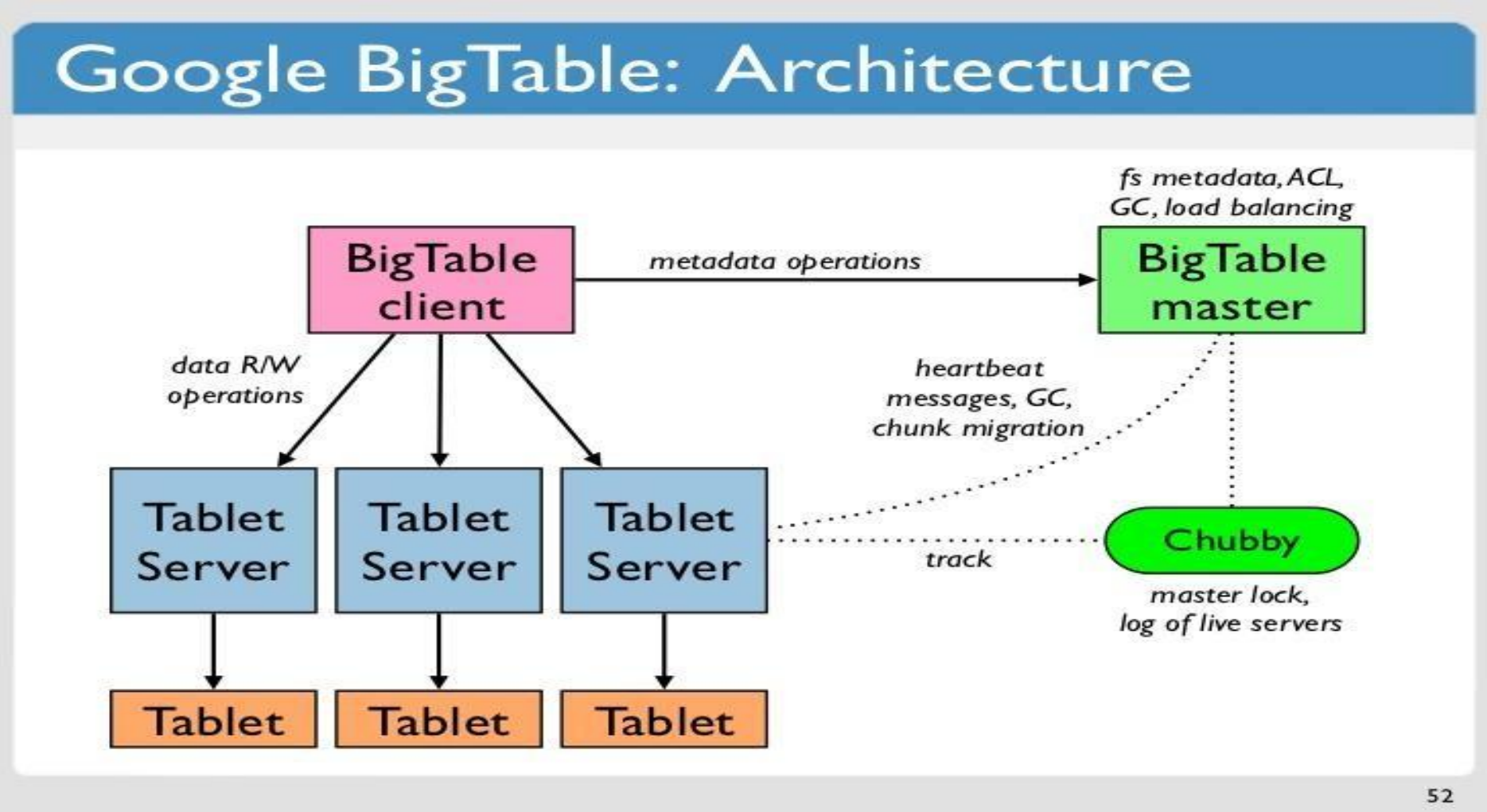
- Escalabilidade:
  - Crescimento horizontal simplificado.
- Não utilizam esquema/restrições referenciais:
  - Maior flexibilidade.
- Processamento Distribuído, de baixo custo.
- Suporte a replicação.

Características

- MAP/REDUCE.



Características



Características

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

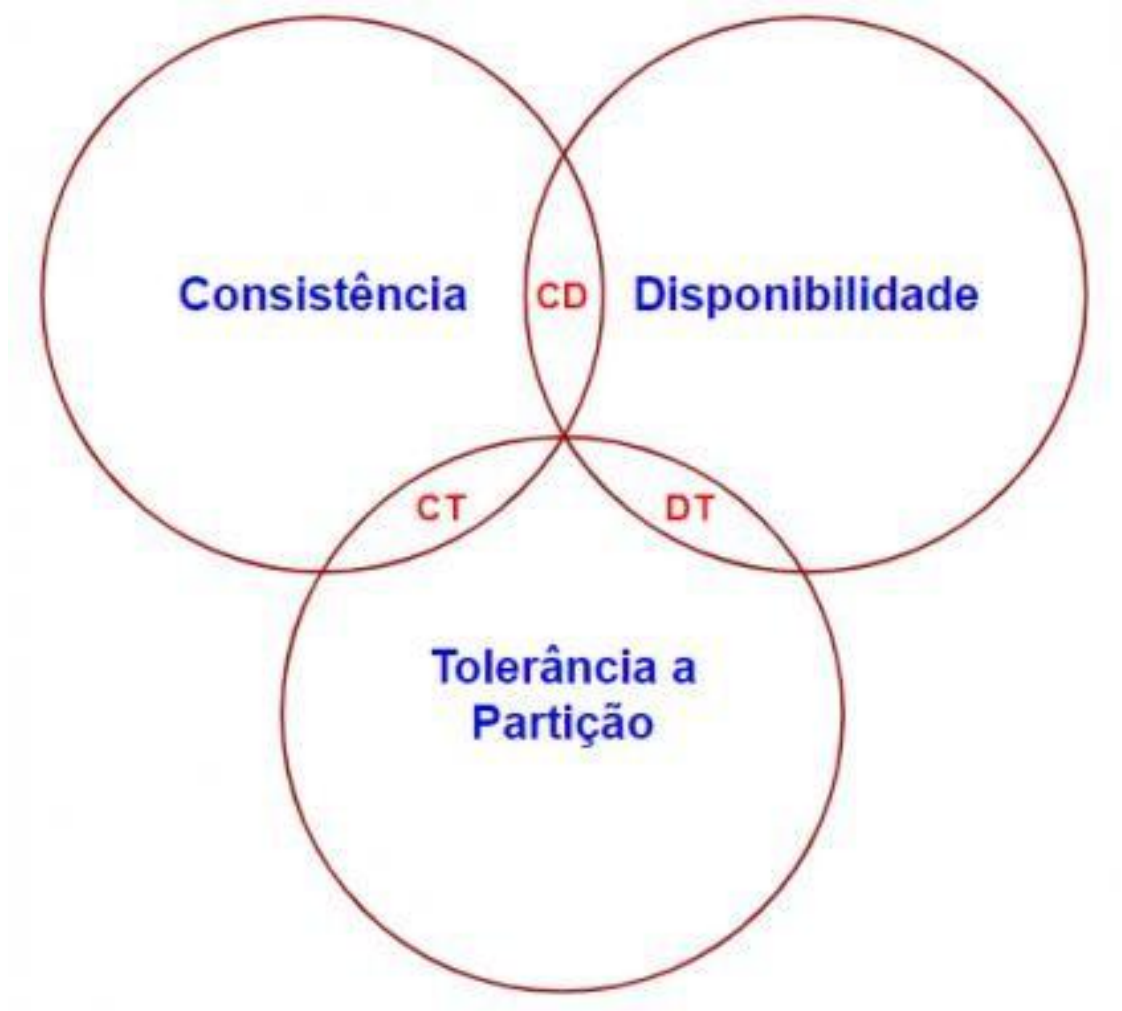
ACID x BASE

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

Teorema CAP

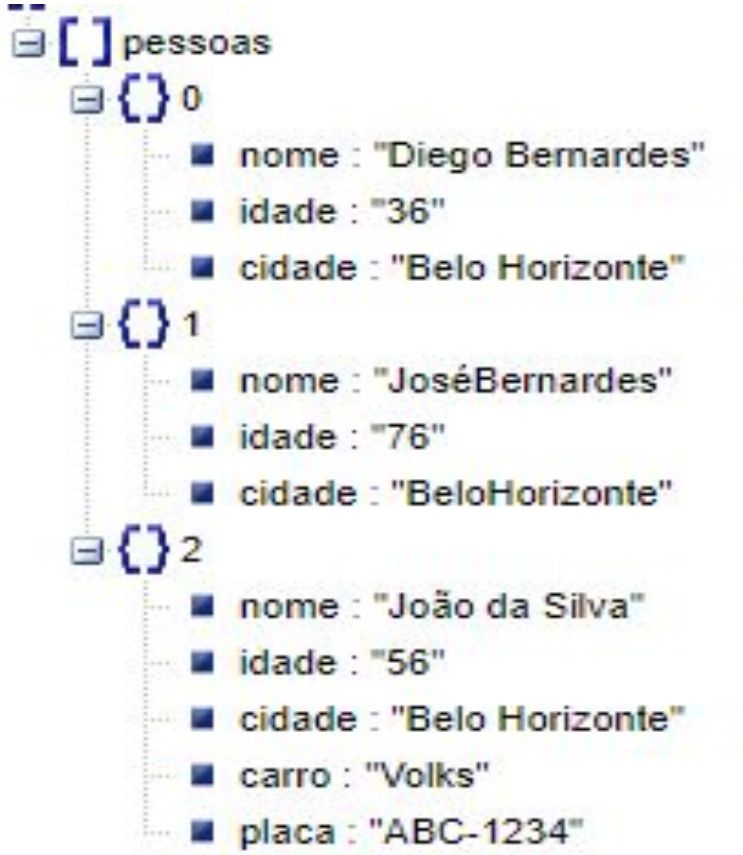
- Uma característica de sistemas de bancos de dados distribuídos é a observação do Teorema CAP:
    - Consistência: Mesma visão dos dados;
    - Disponibilidade: Acesso aos dados sempre disponível;
    - Tolerância à Partição: Manutenção das propriedades independentemente de alterações ou implantações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

Teorema CAP


Consistência
CD
Disponibilidade
CT
DT
Tolerância a
Partição

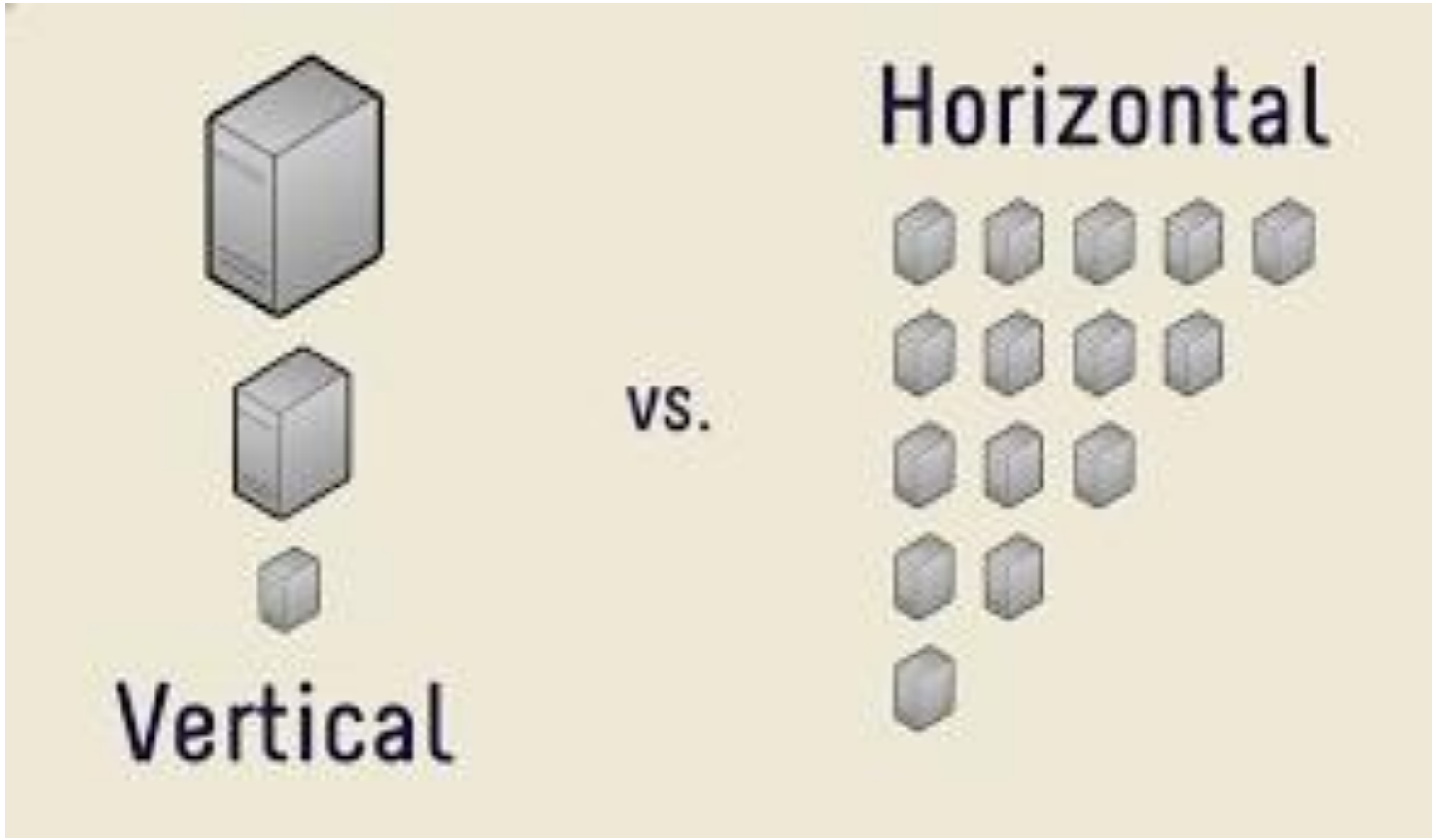
Quando usar NoSQL

- Quando a estrutura de dados é flexível.



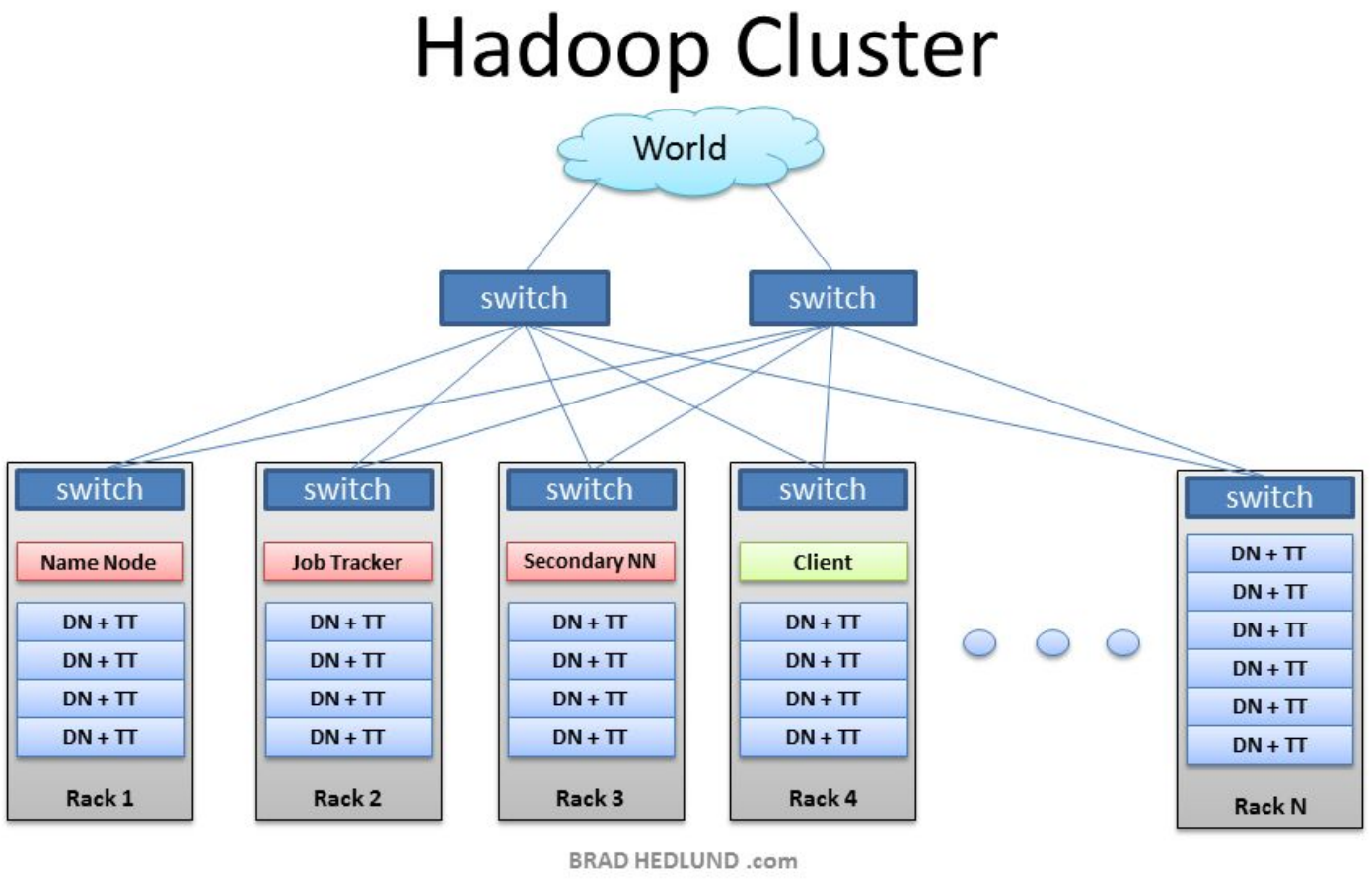
Quando usar NoSQL

- Quando existe a necessidade de escalabilidade ágil e barata.



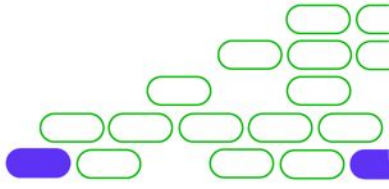
Quando usar NoSQL

- Quando disponibilidade é um requisito crítico.



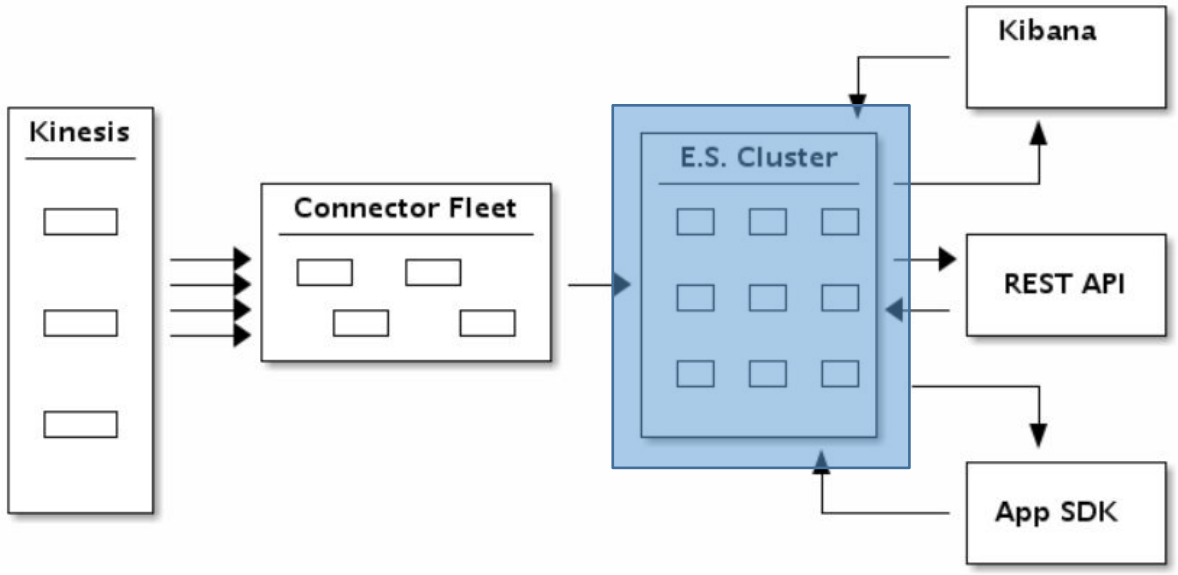
Quando usar NoSQL

- 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.

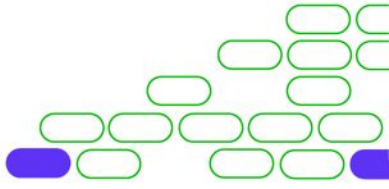


Tipos de Bancos de Dados NoSQL

- 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.

Faculdade XPe

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

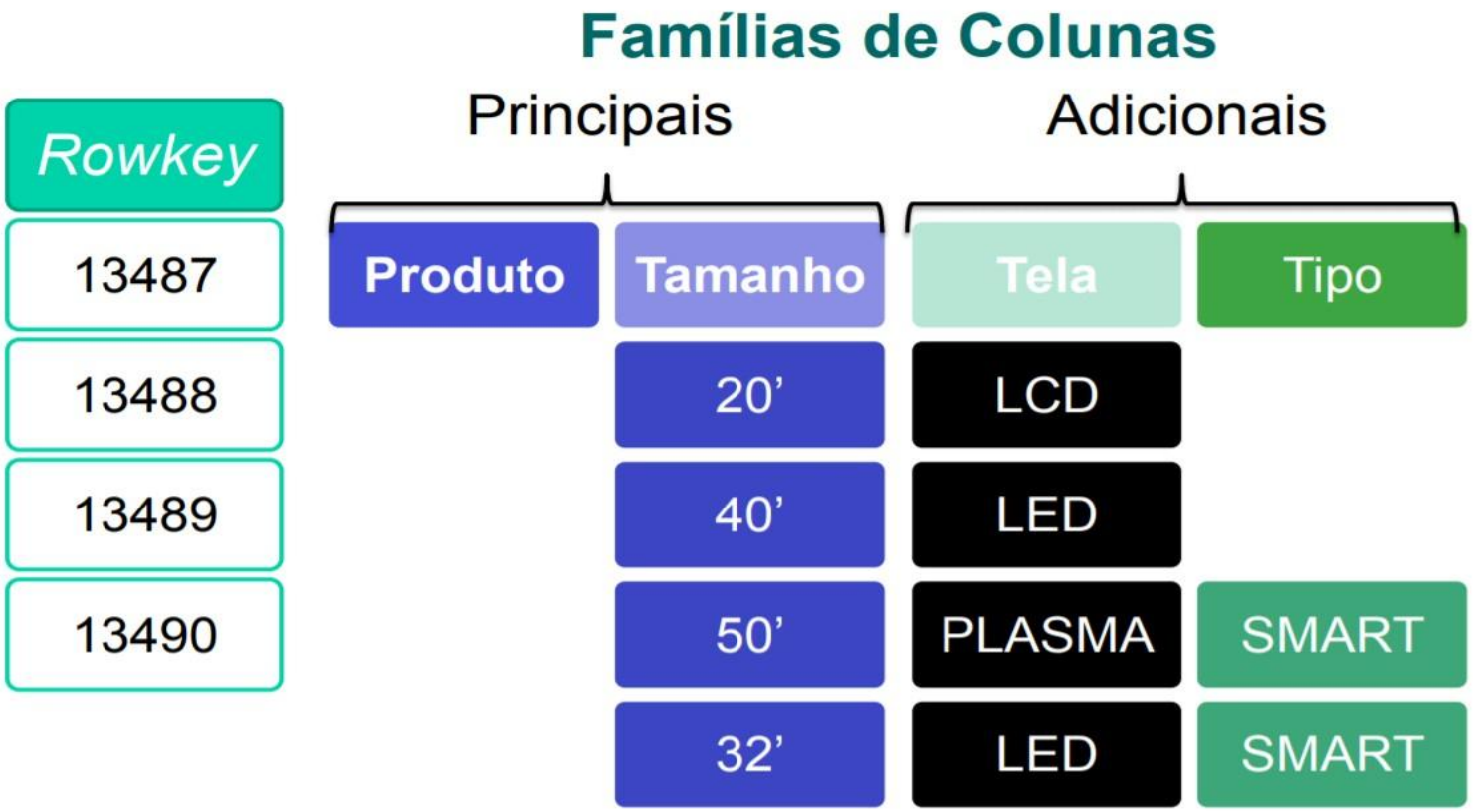
- **Introdução**
  - Um banco de dados colunar é otimizado para recuperação rápida de colunas de dados.
  - Geralmente em aplicações analíticas.
  - O mecanismo colunar reduz o esforço computacional para operações de entrada/saída de dados em disco.
  - O princípio colunar é de trazer apenas os dados 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:**

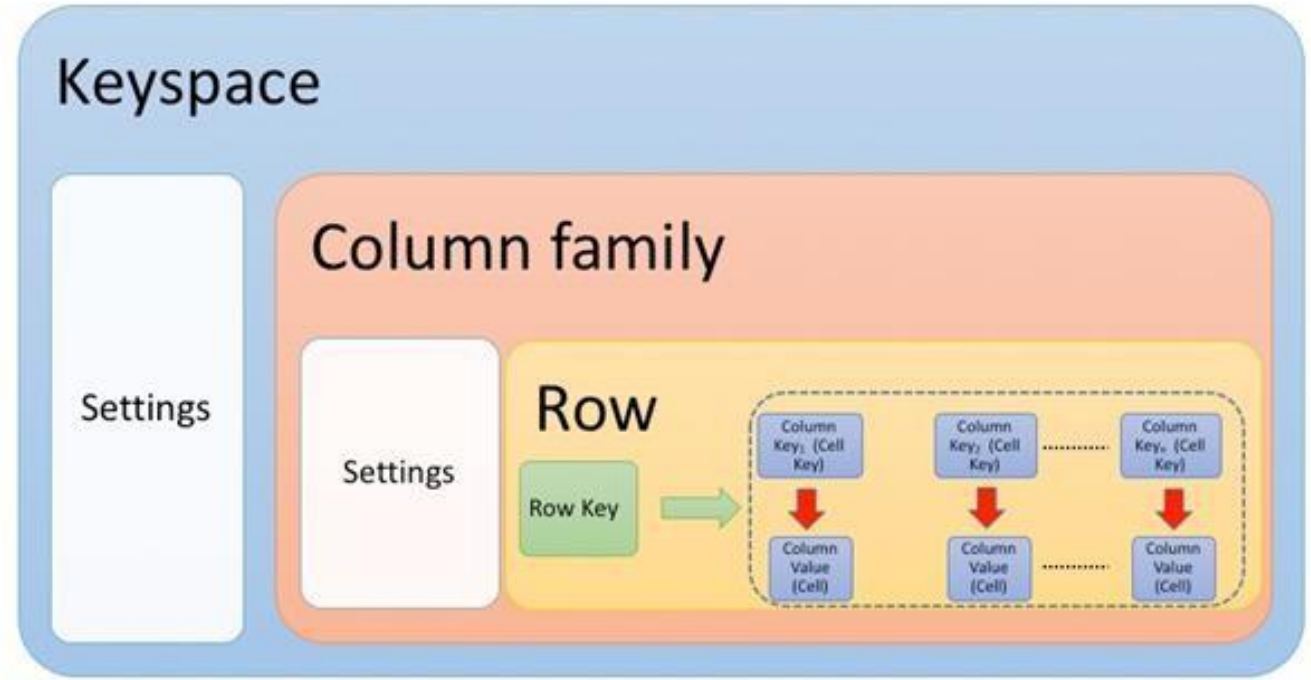


Apache Cassandra

- 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.

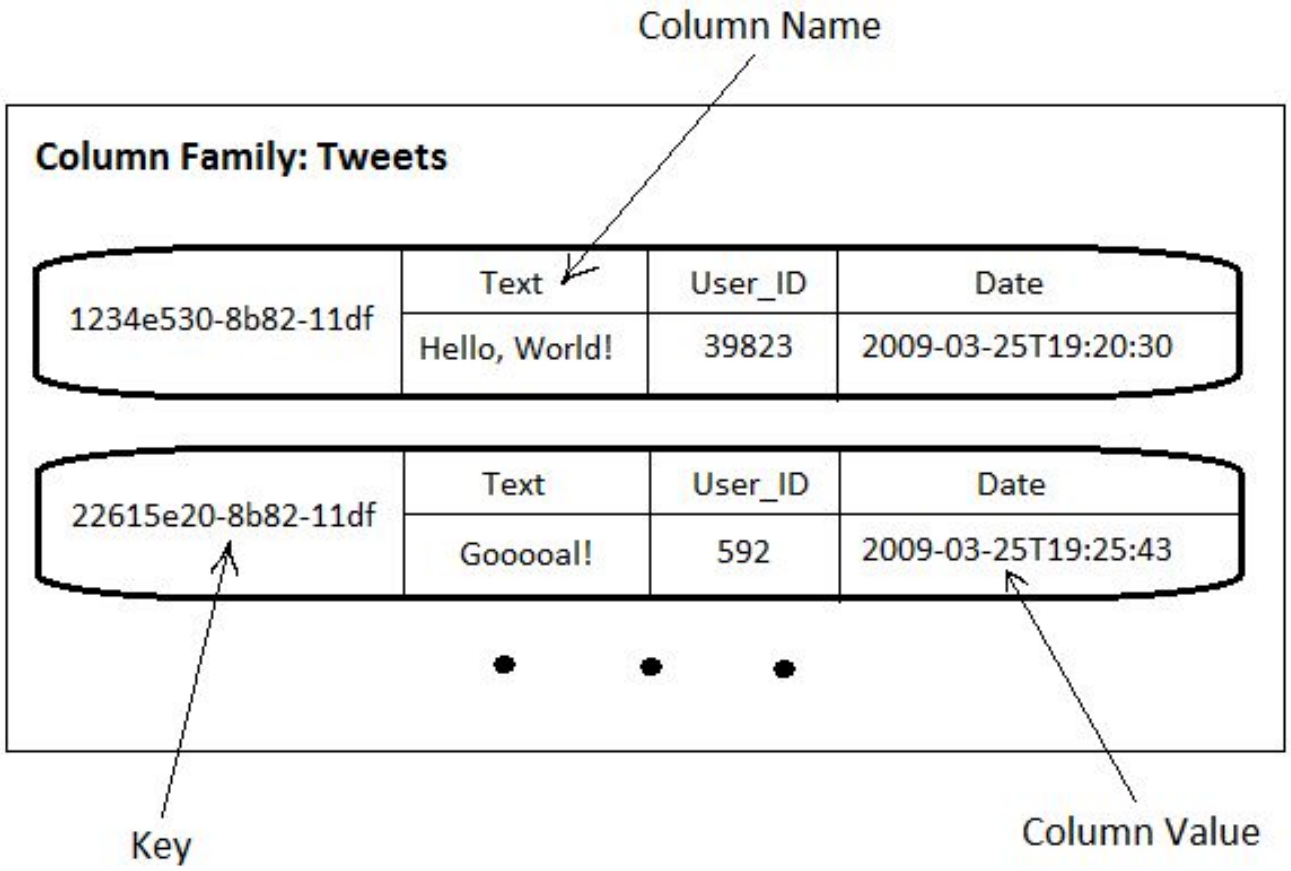
Apache Cassandra

- Arquitetura.



Apache Cassandra

- Modelo de Dados.



Próxima aula...

- Demonstração do Apache Cassandra.

Faculdade XPe

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

Próxima aula...

- Bancos de Dados Colunares.

Faculdade XPe

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:**

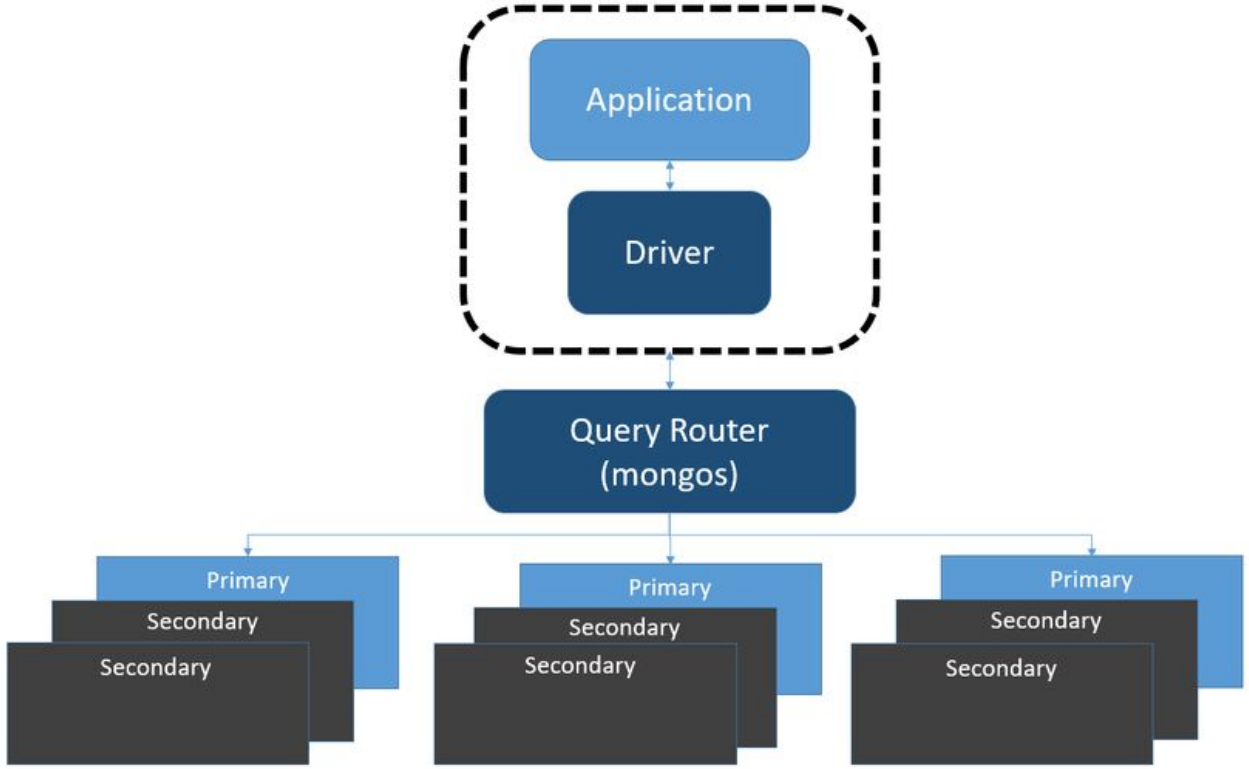
```
{
  "nota_fiscal": [
    {
      "id_compra": 123,
      "valor": 150.00,
      "produto": "Televisão"
    },
    {
      "id_compra": 123,
      "valor": 150.00,
      "produto": "Mesa de Jantar",
      "itens": [
        "cadeira 01",
        "cadeira 02",
        "cadeira 03",
        "cadeira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

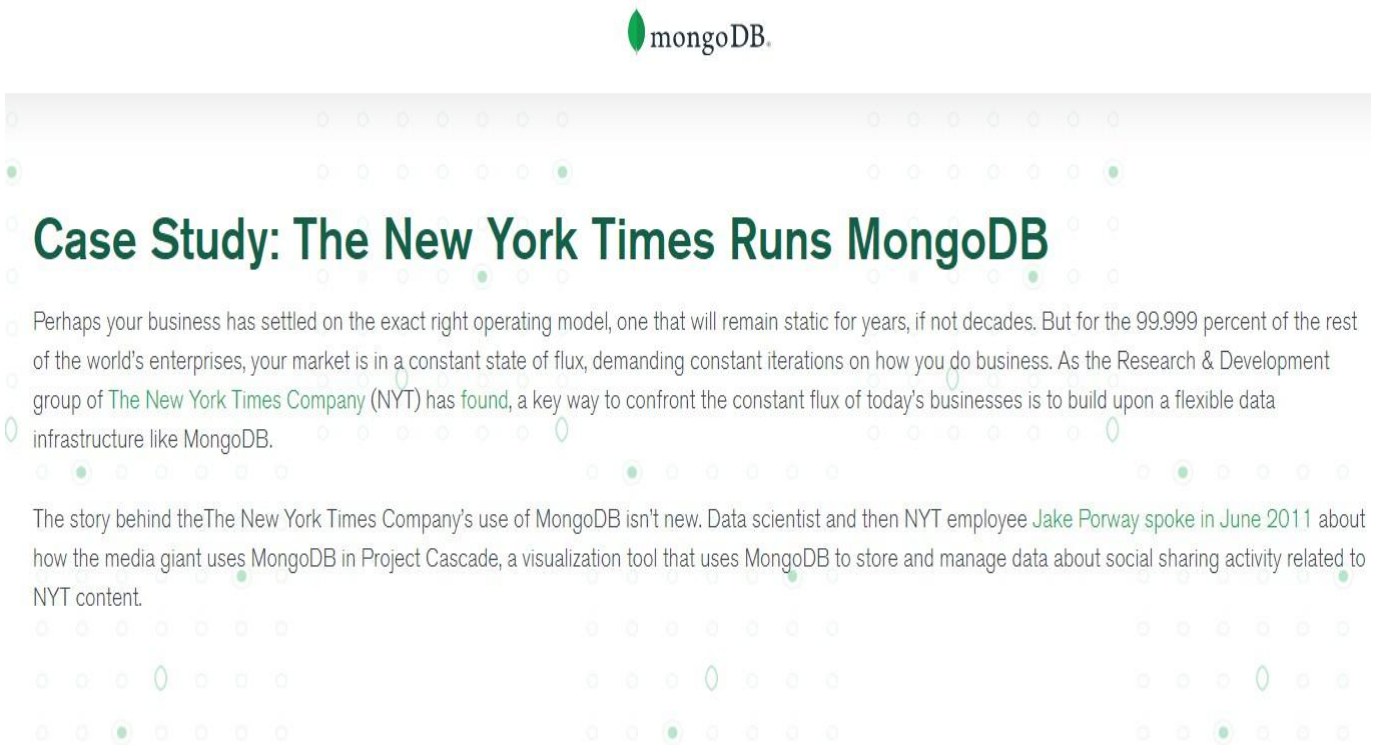
Bancos de Dados de Documentos

- **Arquitetura.**



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

Bancos de Dados de Documentos



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"
        ]
    }
}
```

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

- **Estrutura - JSON**
  - Os dados são separados por vírgulas(,)
  - As chaves {} contém objetos
  - Os colchetes [] expressam matrizes/vetores

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"
      ]
    }
  ]
}
```

Documentos - 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.
- 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

Documentos - Demonstração

- Exemplo de registro no Modelo Relacional

PRIMEIRO_NOME	ULTIMO_NOME	EMAIL	DATA_ADMISSAO	CARGO_ID	SALARIO	DEPARTAMENTO_NOME	CARGO_NOME	CIDADE	PAIS_NOME	REGIAO_NOME
Steven	King	SKING	17/06/2003	AD_PRES	24000.00	Executive	President	Seattle	United States of America	Americas

Consulta SQL

```
SELECT E.[PRIMEIRO_NOME]
    ,E.[ULTIMO_NOME]
    ,E.[EMAIL]
    ,E.[DATA_ADMISSAO]
    ,E.[CARGO_ID]
    ,E.[SALARIO]
        ,D.DEPARTAMENTO_NOME
        ,C.CARGO_NOME
        ,L.CIDADE
        ,P.PAIS_NOME
        ,R.REGIAO_NOME
FROM EMPREGADOS E
INNER JOIN DEPARTAMENTOS D ON E.DEPARTAMENTO_ID = D.DEPARTAMENTO_ID
INNER JOIN CARGOS C ON E.CARGO_ID = C.CARGO_ID
INNER JOIN LOCALIDADES L ON D.LOCALIDADE_ID = L.LOCALIDADE_ID
INNER JOIN PAISES P ON L.PAIS_ID = P.PAIS_ID
INNER JOIN REGIOES R ON P.REGIAO_ID = R.REGIAO_ID
```

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

- **Estrutura - JSON**
  - Os dados são separados por vírgulas(,)
  - As chaves {} contém objetos
  - Os colchetes [] expressam matrizes/vetores

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"
      ]
    }
  ]
}
```

Documentos - 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.
- 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

Documentos - Demonstração

- Exemplo de registro no Modelo Relacional

PRIMEIRO_NOME	ULTIMO_NOME	EMAIL	DATA_ADMISSAO	CARGO_ID	SALARIO	DEPARTAMENTO_NOME	CARGO_NOME	CIDADE	PAIS_NOME	REGIAO_NOME
Steven	King	SKING	17/06/2003	AD_PRES	24000.00	Executive	President	Seattle	United States of America	Americas

Consulta SQL

```
SELECT E.[PRIMEIRO_NOME]
    ,E.[ULTIMO_NOME]
    ,E.[EMAIL]
    ,E.[DATA_ADMISSAO]
    ,E.[CARGO_ID]
    ,E.[SALARIO]
        ,D.DEPARTAMENTO_NOME
        ,C.CARGO_NOME
        ,L.CIDADE
        ,P.PAIS_NOME
        ,R.REGIAO_NOME
FROM EMPREGADOS E
INNER JOIN DEPARTAMENTOS D ON E.DEPARTAMENTO_ID = D.DEPARTAMENTO_ID
INNER JOIN CARGOS C ON E.CARGO_ID = C.CARGO_ID
INNER JOIN LOCALIDADES L ON D.LOCALIDADE_ID = L.LOCALIDADE_ID
INNER JOIN PAISES P ON L.PAIS_ID = P.PAIS_ID
INNER JOIN REGIOES R ON P.REGIAO_ID = R.REGIAO_ID
```

Próxima aula...

- Demonstração do MongoDB.

Faculdade XPe

Fundamentos de Bancos de Dados

Capítulo 04. Conceitos Complementares

Prof. Diego Bernardes

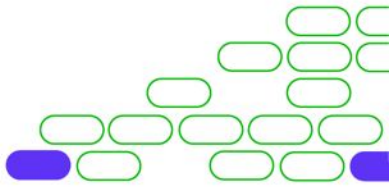
Faculdade XPe

Fundamentos de Bancos de Dados

Capítulo 05. Conceitos Complementares

Prof. Diego Bernardes

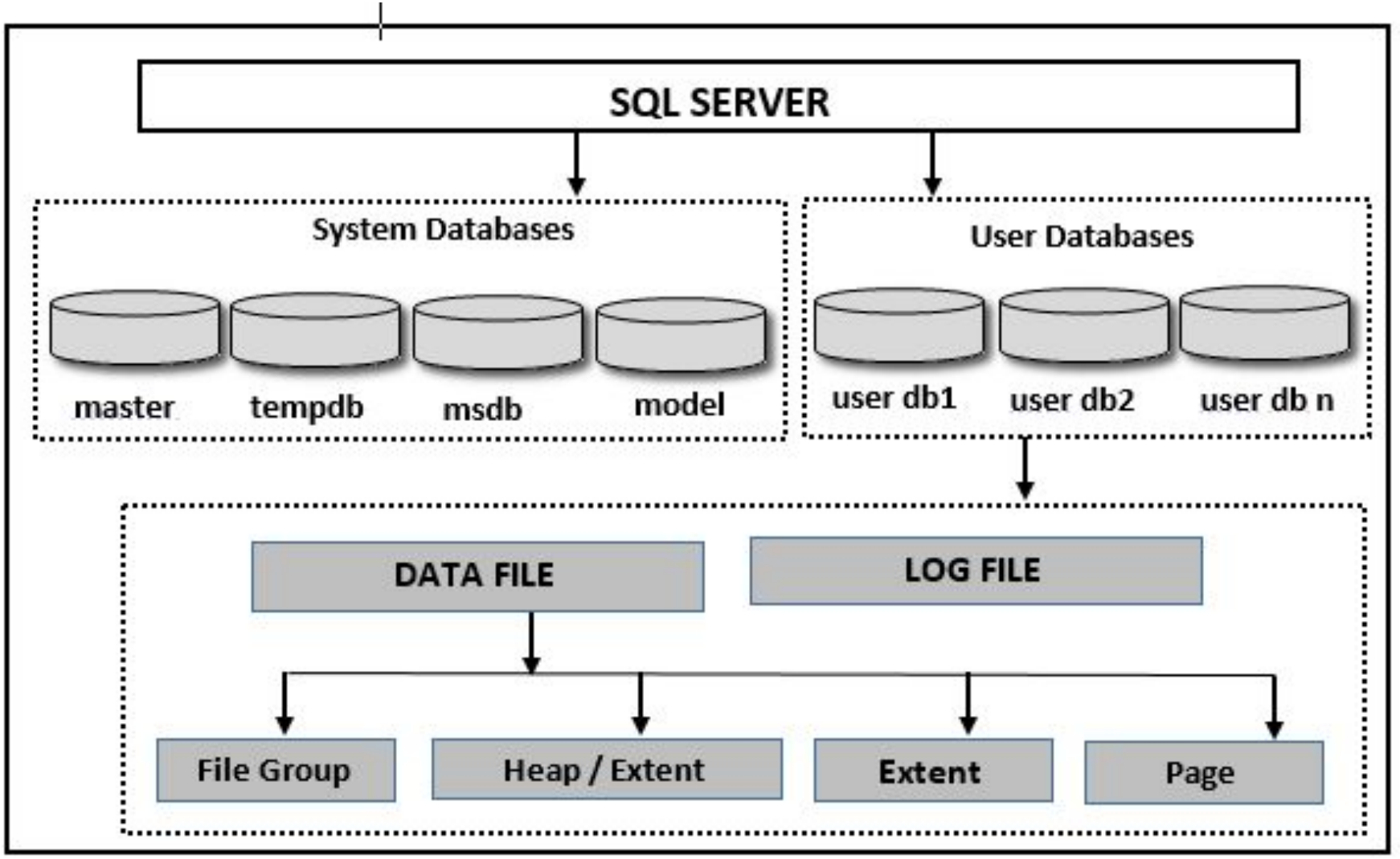
- Introdução ao SGBD SQL Server



Introdução

- O Microsoft SQL Server é o SGBD desenvolvido pela Microsoft para o Mercado de Bancos de Dados Relacionais.
- Possui componentes de bases transacionais e analíticas.
- Possui mecanismos de alta disponibilidade
- Possui integração com outras ferramentas da Microsoft
- Possui a linguagem T-SQL para desenvolvimento

Arquitetura



Faculdade XPe

Fundamentos de Bancos de Dados

Capítulo 04. Conceitos Complementares

Prof. Diego Bernardes

Faculdade XPe

Fundamentos de Bancos de Dados

Capítulo 04. Conceitos Complementares

Prof. Diego Bernardes

Nesta Aula

- Introdução aos Conceitos de Normalização

Introdução

- Normalização é um processo de otimização de modelo de dados, com os seguintes objetivos:
  - Eliminar redundância de dados
  - Eliminar inconsistências de dados
  - Trazer coesão para as tabelas do banco de dados – Concentração de assuntos

Normalização

- Ganhos de Modelagem
  - Independência dos dados
  - Minimizar riscos redundâncias, que por sua vez minimiza os riscos de inconsistências
  - Facilitar a manipulação do Banco de Dados
  - Facilitar a manutenção dos sistemas de informação, sem grandes impactos.

Formas Normais

- Conjunto de regras aplicadas para atingir a normalização de um modelo de dados.
- Existem 5 Formas Normais, porém a aplicação das 3 primeiras formas normais resolvem praticamente todos os tipos de modelos.

Formas Normais

- 1FN

- Todos os atributos de uma tabela devem ser atômicos, ou seja, a tabela não deve conter grupos repetidos e nem atributos com mais de um valor.

codigo	nome	Telefone
1	Antônio Silva	12345678 87654321
2	Helena Abreu	12341234

codigo	nome
1	Antônio Silva
2	Helena Abreu

codigo	numero	cod_pessoa
1	12345678	1
2	87654321	1

Formas Normais

- 2FN

- Antes de mais nada, para estar na 2FN é preciso estar na 1FN.

- Além disso, todos os atributos não chaves da tabela devem depender unicamente da chave primária.

matricula	aluno	curso
1	José Bernardes	Banco de Dados
2	Joana Silveira	Sistemas de Informação

matricula	aluno	cod_curso
1	José Bernardes	1
2	Joana Silveira	2

cod_curso	nome_curso
1	Banco de Dados
2	Sistemas de Informação

Formas Normais

- 3FN

- Para estar na 3FN, é preciso estar na 2FN. Além disso, os atributos não chave de uma tabela devem ser mutuamente independentes

matricula	aluno	curso
1	José Bernardes	Banco de Dados
2	Joana Silveira	Sistemas de Informação

matricula	aluno
1	José Bernardes
2	Joana Silveira

Cod_curso	matricula_aluno
1	1
2	2

cod_curso	nome_curso
1	Banco de Dados
2	Sistemas de Informação

Demonstração

Faculdade XPe

Fundamentos de Bancos de Dados

Capítulo 04. Conceitos Complementares

Prof. Diego Bernardes

Faculdade XPe

Fundamentos de Bancos de Dados

Capítulo 04. Conceitos Complementares

Prof. Diego Bernardes

Nesta Aula

- Boas práticas de modelagem relacional

Introdução

- Os modelos de dados discutidos até aqui atendem às regras de Projetos de Bancos de Dados
- Mas, para evolução contínua, muitas empresas optam por definir padrões e melhorias nos modelos desde sua origem.
- Via de regra, esses padrões são documentados em documentos de Padrões Corporativos de Bancos de Dados.
- Muitas vezes, os modelos passam por essas equipes e são reprovados caso não sejam adequados aos padrões.

Algumas Sugestões

- Padrão de nomenclatura (prefixos, sufixos)
- Campos Nulos / Não nulos
- Tipos de dados com precisão
- Nomes de chaves
- Criação de indices / únicos / cluster

Demonstração

Faculdade XPe

Fundamentos de Bancos de Dados

Capítulo 04. Conceitos Complementares

Prof. Diego Bernardes

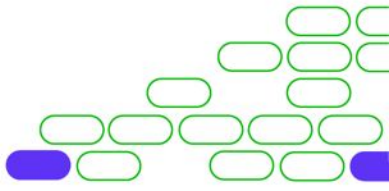
Faculdade XPe

Fundamentos de Bancos de Dados

Capítulo 04. Conceitos Complementares

Prof. Diego Bernardes

- Introdução aos Bancos de Dados de Grafos



Introdução aos Grafos

- **Introdução**

  - Muitos problemas de otimização podem ser analisados utilizando-se uma estrutura denominada grafo ou rede.

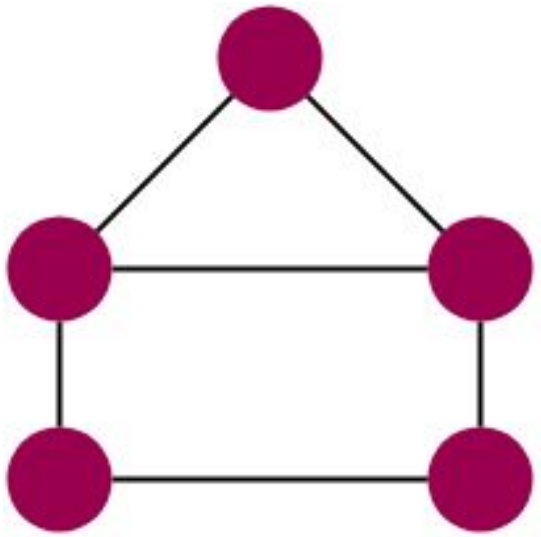
  - Esse tipo de estrutura é utilizado em problemas de computação para modelagem de relacionamentos entre estruturas de dados, sem hierarquia.

  - Os relacionamentos podem ou não possuir direção, representar circuitos, caminhos ou rotas.

Grafos

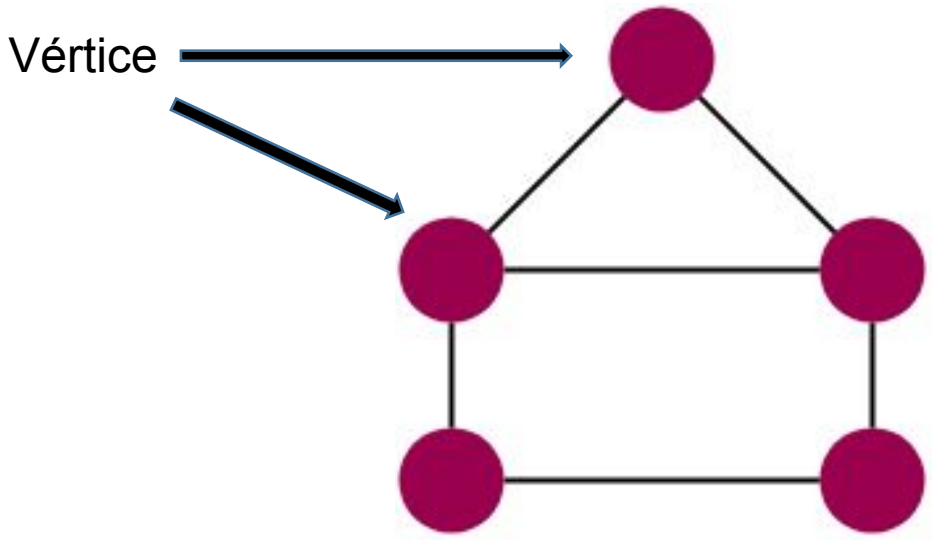
- **Definição**

Grafo - O grafo G=(N,E) consiste em um conjunto de nós denotados por N, ou por N(G) e conjunto de arestas E ou E(G).



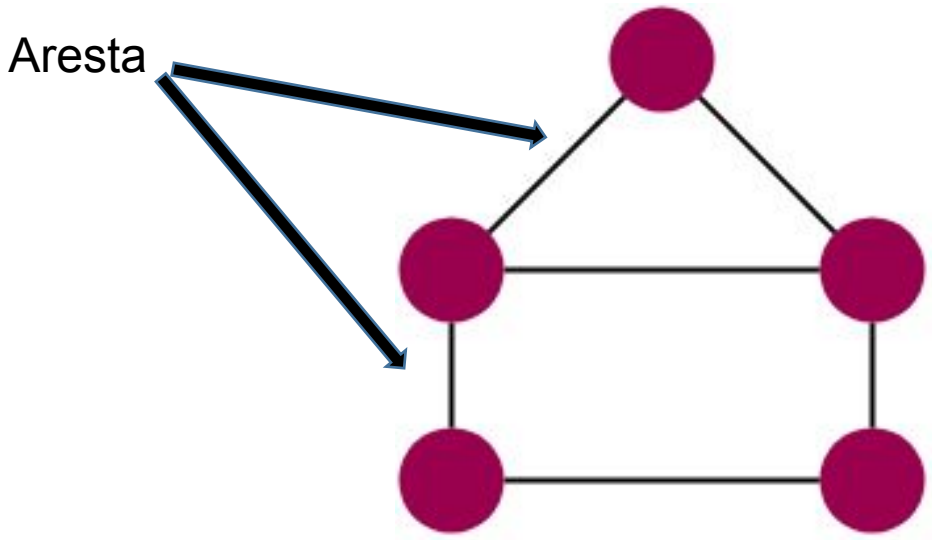
Grafos

- **Definição – Vértice**
    - São nós ou pontos da rede que são as estruturas principais do grafo, que teoricamente é a estrutura de armazenamento.



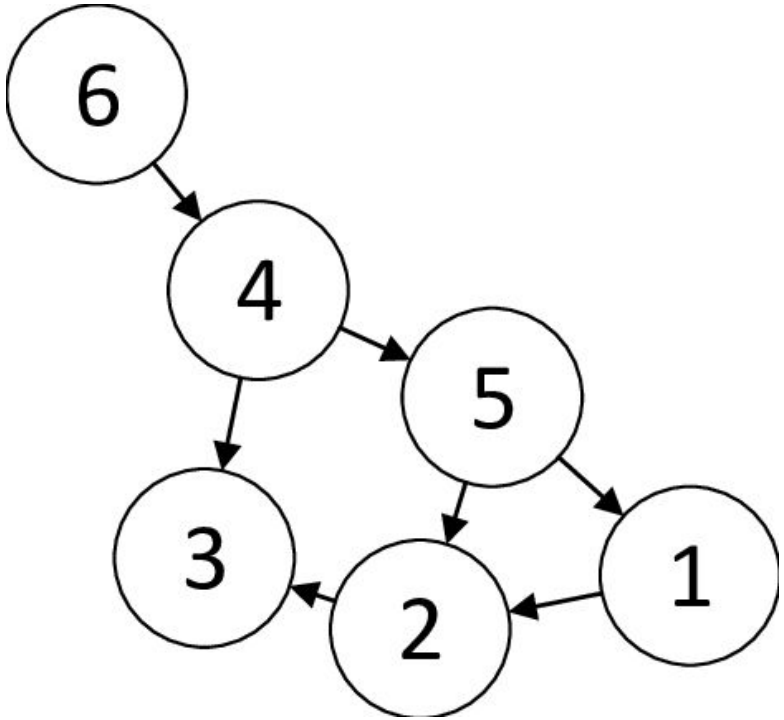
Grafos

- **Definição – Aresta**
    - São as conexões entre as arestas, em uma estrutura de rede são as ligações diretas entre dois pontos.



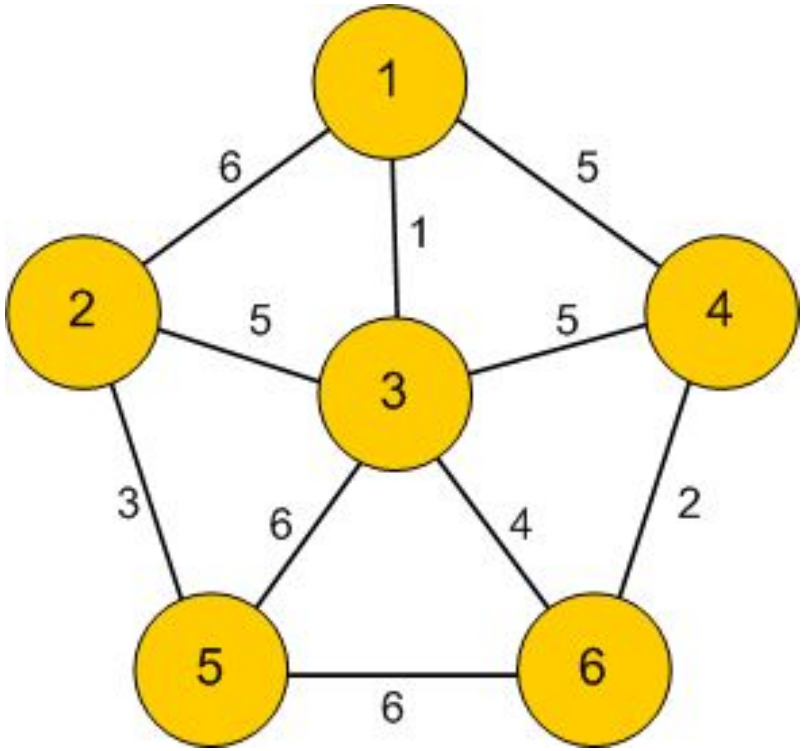
Grafos Dirigidos

- As relações possuem direção, ou seja, o relacionamento sai de um nó A para um nó B, através de uma ligação representada por uma seta.



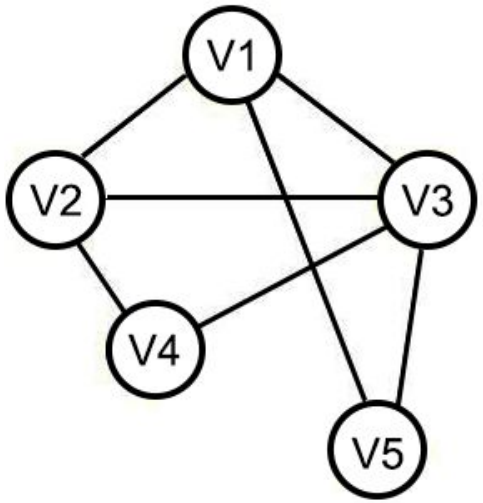
Arestas com Peso

- Peso em arestas são informações que indicam a “força” ou “custo” na relação de um nó com outro, em relação aos demais.



Matriz de Adjacências

- Uma forma comum de representação computacional de um Grafo



	V1	V2	V3	V4	V5
V1	0	1	1	0	1
V2	1	0	1	1	0
V3	1	1	0	1	1
V4	0	1	1	0	0
V5	1	0	1	0	0

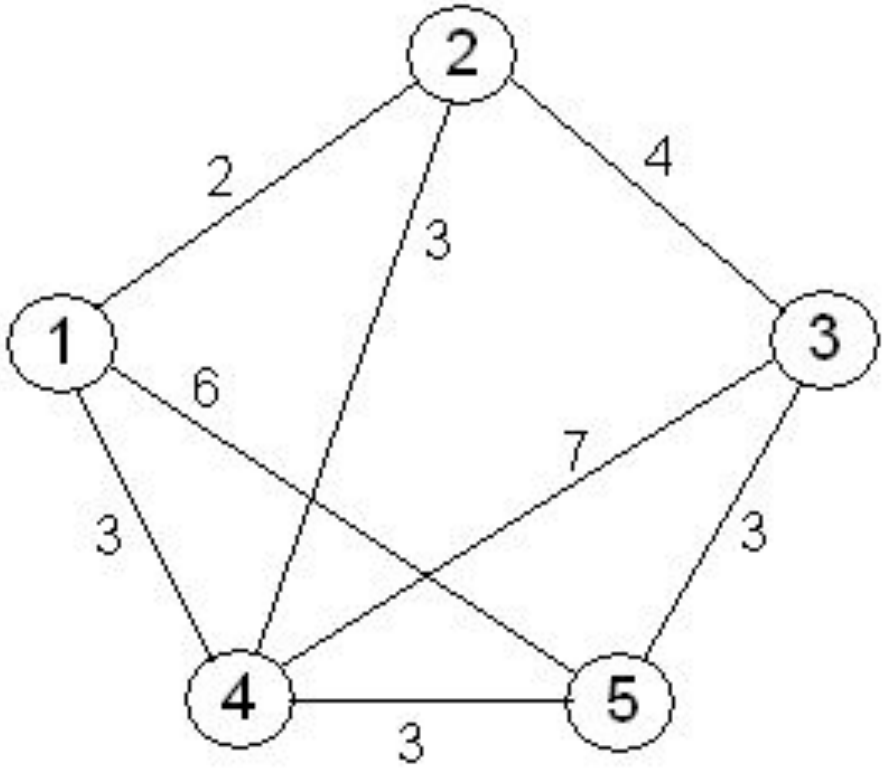
Fonte: Paulo Martins (medium.com)

Problema do Caixeiro Viajante

- Um dos problemas clássicos da Teoria de Grafos em Ciência da Computação.

Problema do Caixeiro Viajante

- Exemplo de Modelagem:



Próxima Aula...

- Introdução às Bases de Dados Orientadas a Grafos

Faculdade XPe

Fundamentos de Bancos de Dados

Capítulo 04. Conceitos Complementares

Prof. Diego Bernardes

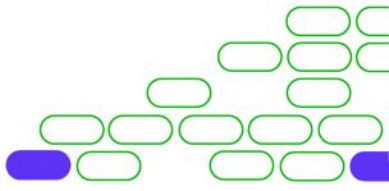
Faculdade XPe

Fundamentos de Bancos de Dados

Capítulo 04. Conceitos Complementares

Prof. Diego Bernardes

- Introdução aos Bancos de Dados de Grafos



Bancos de Dados de Grafos

- **Aplicações**

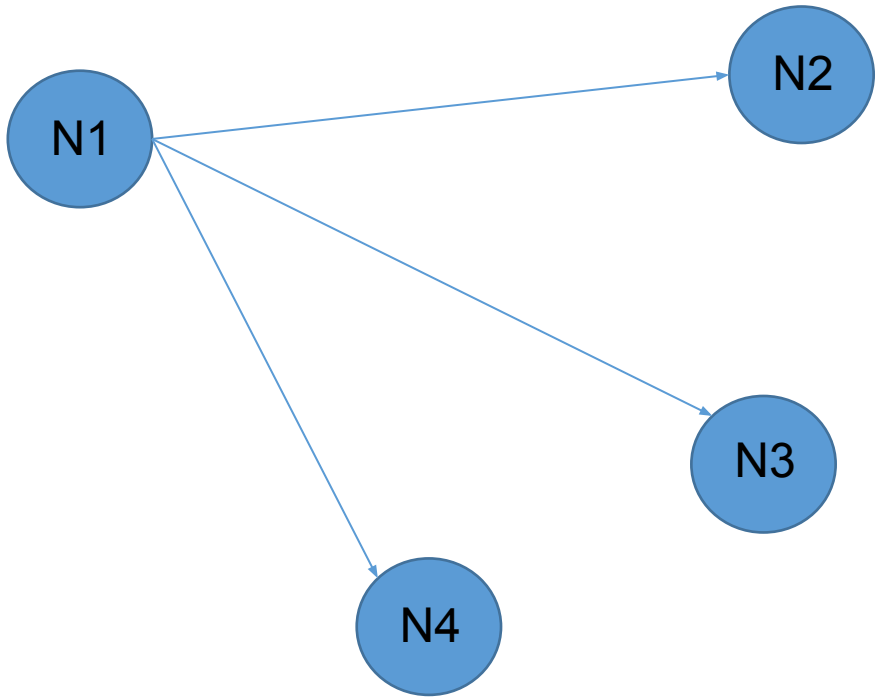
  - Web Semântica
  - Redes de Computadores
  - Mecanismos de Recomendação
  - Informações que possuem relacionamentos de dependência ou de similaridade.

Aplicações

- **Bancos de Dados de Grafos**
  - Neo4J
  - Cypher
- Quando utilizar?
  - É possível modelar “traduzir” bases de dados relacionais para bases de dados de Grafos.
  - O ganho, porém, é quando as relações tem importâncias diferentes e direções.
  - Uma relação 1-N, por exemplo, seria representada por várias arestas.

Introdução aos Grafos

- **Relação 1 – N em um Grafo**



Introdução aos Grafos

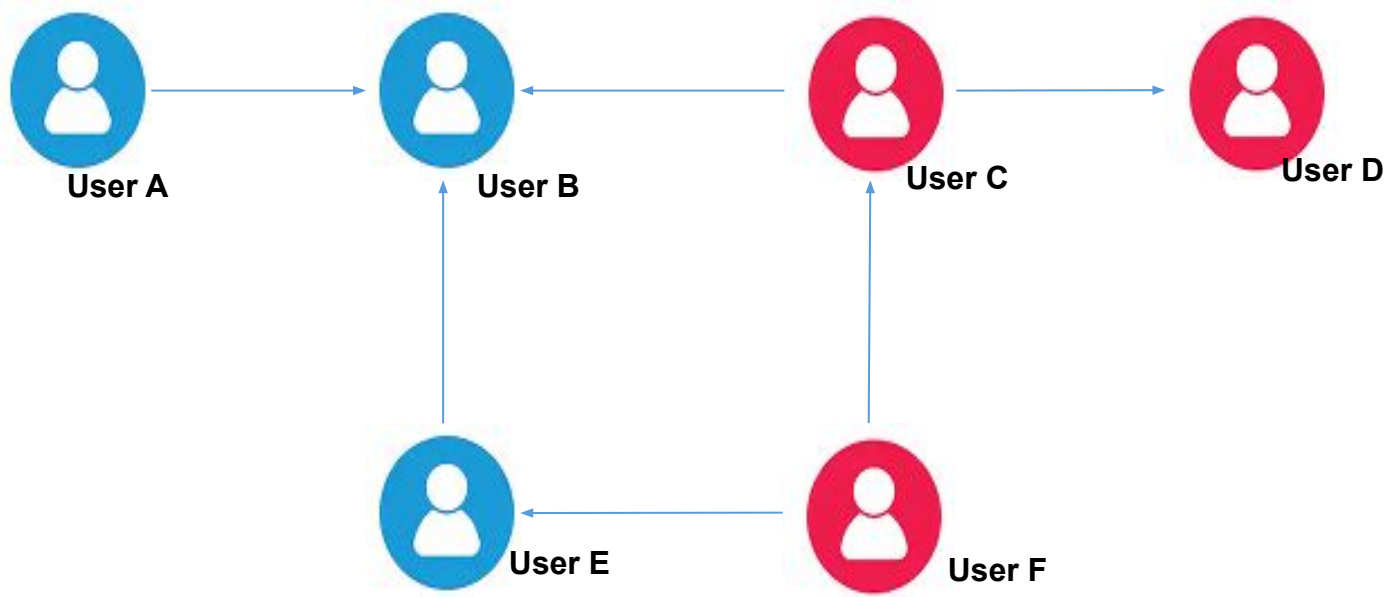
- **Considere o seguinte exemplo (apenas para entendimento da visão de Grafos)**
  - Considere a rede social Instagram
  - Vamos extrair um “recorte” dessa rede, para compreender como algumas pessoas se relacionam.
  - Em nossa rede, temos os seguintes usuários:
    - User A
    - User B
    - User C
    - User D
    - User E
    - User F

Introdução aos Grafos

- Os usuários se relacionam da seguinte maneira
    - User A segue o User B
    - User B não segue ninguém
    - User C segue o User B e User D
    - User D não segue ninguém
    - User E segue o user B
    - User F segue o User B e o User F

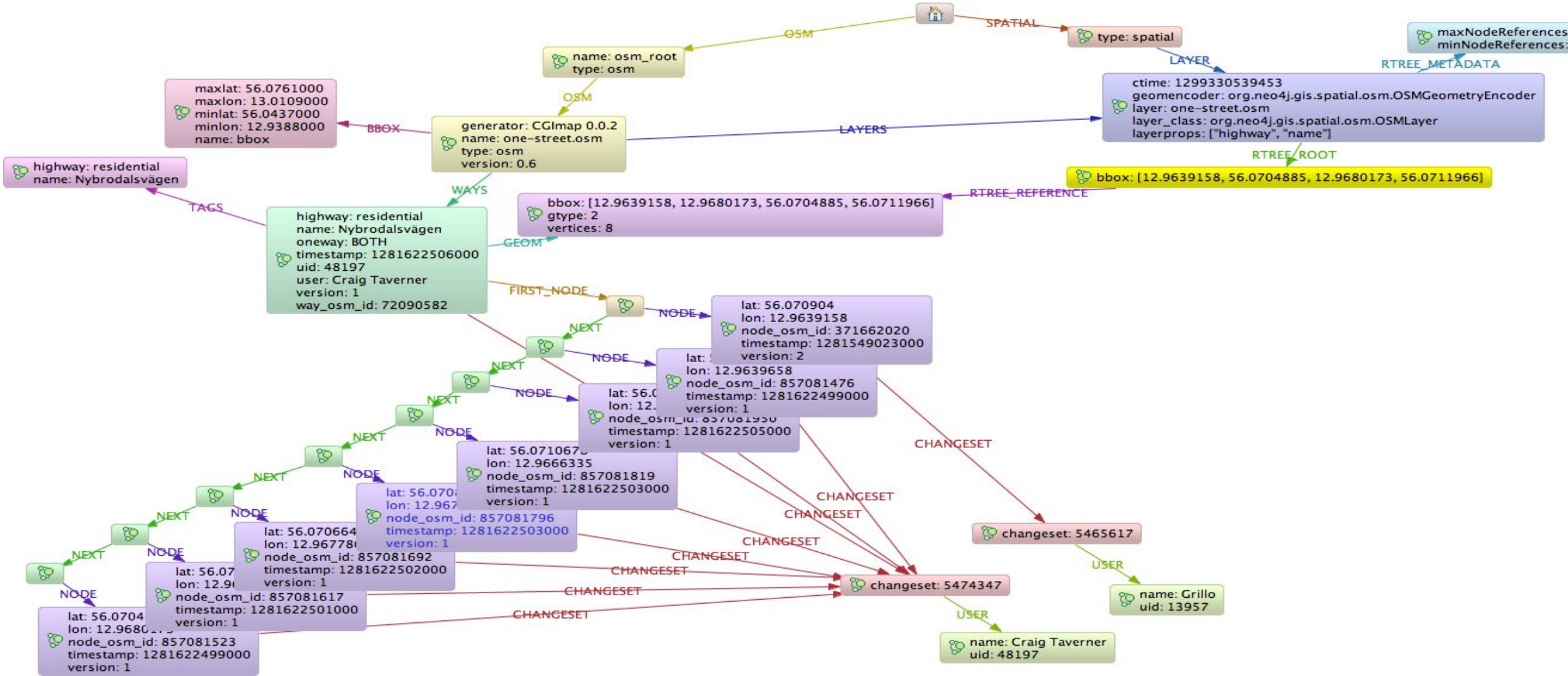
Introdução aos Grafos

- Graficamente falando...

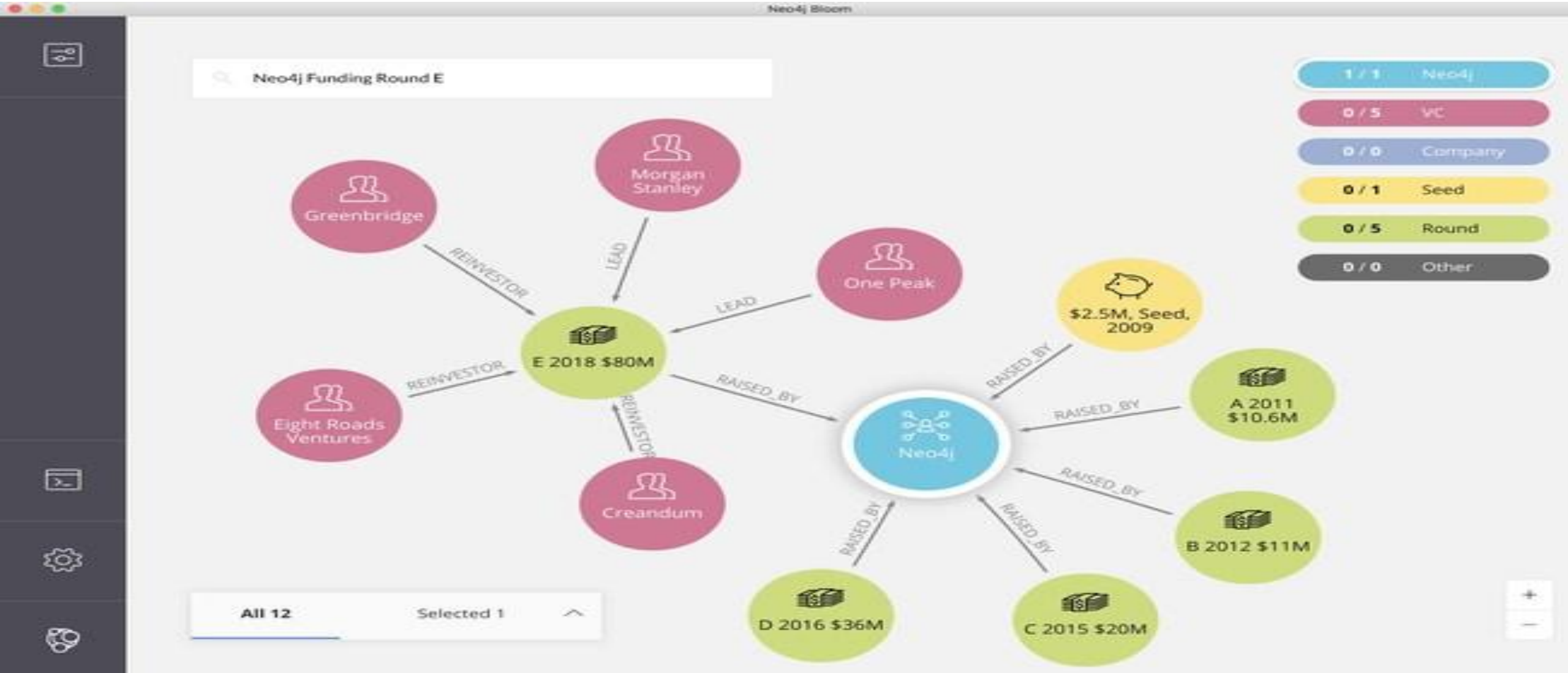


- Na prática, BDs Orientados a Grafos armazenam centenas ou milhares de grafos, como exibido acima.

Neo4J


OSM
SPATIAL
name: osm_root
type: osm
type: spatial
maxNodeReferences: 100
minNodeReferences: 51
LAYER
RTREE_METADATA
maxlat: 56.0761000
maxlon: 13.0109000
minlat: 56.0437000
minlon: 12.9388000
name: bbox
BBOX
OSM
generator: CGImap 0.0.2
name: one-street.osm
type: osm
version: 0.6
LAYERS
ctime: 1299330539453
geomencoder: org.neo4j.gis.spatial.osm.OSMGeometryEncoder
layer: one-street.osm
layer_class: org.neo4j.gis.spatial.osm.OSMLayer
layerprops: ["highway", "name"]
RTREE_ROOT
highway: residential
name: Nybrodalsvägen
TAGS
WAYS
bbox: [12.9639158, 56.0704885, 12.9680173, 56.0711966]
RTREE_REFERENCE
bbox: [12.9639158, 12.9680173, 56.0704885, 56.0711966]
gtype: 2
vertices: 8
GEOM
highway: residential
name: Nybrodalsvägen
oneway: BOTH
timestamp: 1281622506000
uid: 48197
user: Craig Taverner
version: 1
way_osm_id: 72090582
FIRST_NODE
NODE
NEXT
lat: 56.070904
lon: 12.9639158
node_osm_id: 371662020
timestamp: 1281549023000
version: 2
lon: 12.9639658
node_osm_id: 857081476
timestamp: 1281622499000
version: 1
timestamp: 1281622505000
lon: 12.9666335
node_osm_id: 857081819
timestamp: 1281622503000
node_osm_id: 857081796
timestamp: 1281622503000
node_osm_id: 857081692
timestamp: 1281622502000
node_osm_id: 857081617
timestamp: 1281622501000
node_osm_id: 857081523
timestamp: 1281622499000
CHANGESET
changeset: 5465617
changeset: 5474347
USER
name: Grillo
uid: 13957
name: Craig Taverner
uid: 48197

Neo4J


Neo4j Bloom
Neo4j Funding Round E
1 / 1 Neo4j
0 / 5 VC
0 / 0 Company
0 / 1 Seed
0 / 5 Round
0 / 0 Other
Morgan Stanley
Greenbridge
One Peak
$2.5M, Seed, 2009
E 2018 $80M
Eight Roads Ventures
Creandum
Neo4j
A 2011 $10.6M
B 2012 $11M
C 2015 $20M
D 2016 $36M
REINVESTOR
LEAD
RAISED_BY
All 12
Selected 1

Próxima Aula...

- Modelagem de Dados – Neo4J

Faculdade XPe

Fundamentos de Bancos de Dados

Capítulo 04. Conceitos Complementares

Prof. Diego Bernardes

Faculdade XPe

Fundamentos de Bancos de Dados

Capítulo 04. Conceitos Complementares

Prof. Diego Bernardes

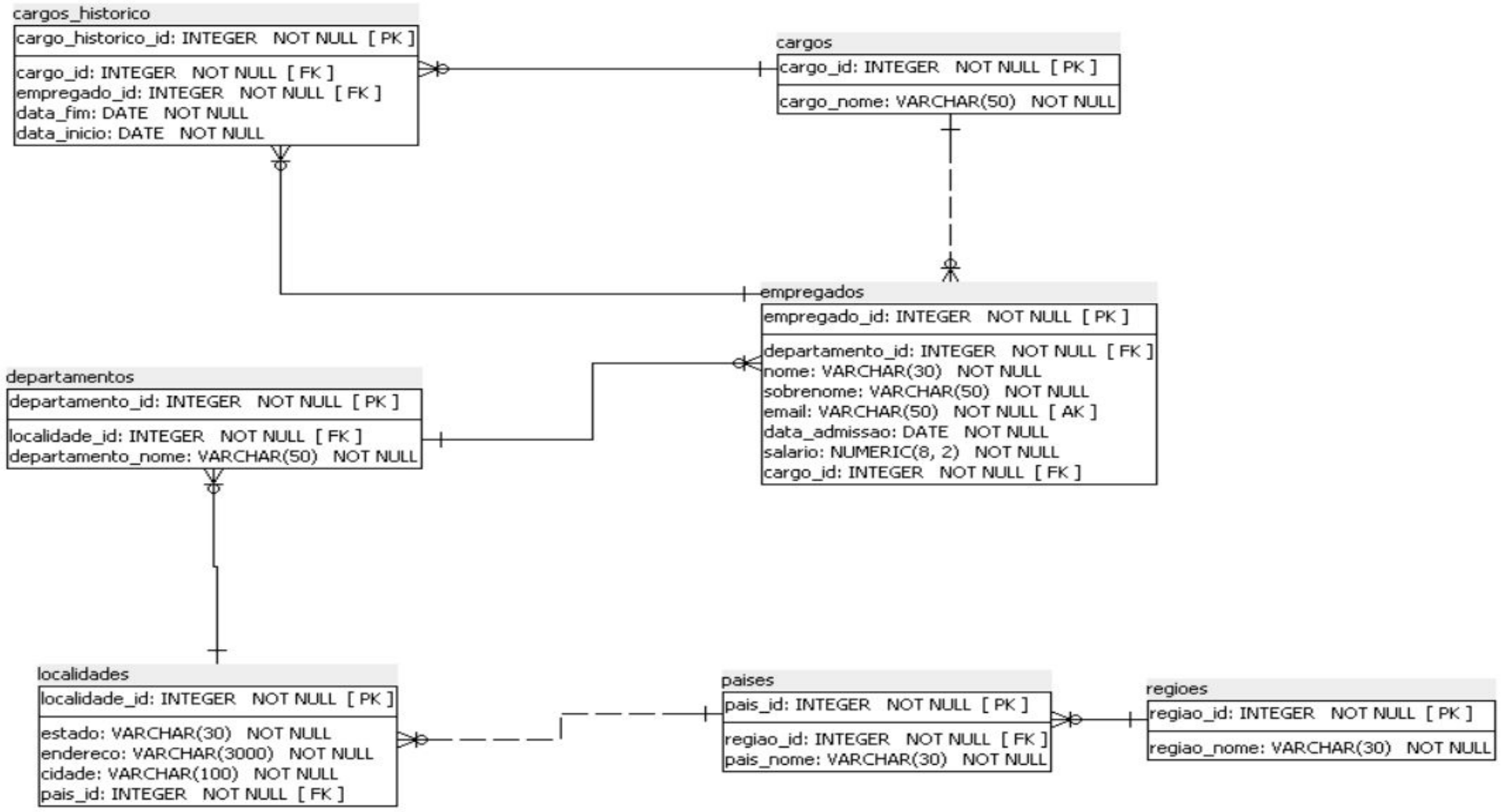
Nesta Aula

- Tópicos sobre desnormalização

Introdução

- Bancos de dados relacionais normalmente devem ser normalizados para eliminação de redundância e inconsistência.
- Existem, porém, ambientes onde remover as Formas Normais se faz necessário em razão de questões de desempenho.
- A seguir vamos falar sobre esse tema e exemplificar.

Nosso Modelo



Na Prática

empregado_id	nome	email	dt_admissão	salario	departamento_id
1	Alexandre	alexandre@igti.com.br	01/01/1995	5000	1
2	João	joao@igti.com.br	25/05/2000	7000	2
3	Ana	ana@igti.com.br	06/01/2012	6500	3
4	Maria	maria@igti.com.br	12/08/2009	8500	2
5	Fernando	fernando@igti.com.br	05/05/2011	7500	1

departamento_id	nome
1	Vendas
2	T.I
3	RH

Desnormalizando o Modelo

empregado_id	nome	email	dt_admissão	salario	departamento_id	departamento_nome
1	Alexandre	alexandre@igti.com.br	01/01/1995	5000	1	Vendas
2	João	joao@igti.com.br	25/05/2000	7000	2	T.I
3	Ana	ana@igti.com.br	06/01/2012	6500	3	RH
4	Maria	maria@igti.com.br	12/08/2009	8500	2	T.I
5	Fernando	fernando@igti.com.br	05/05/2011	7500	1	Vendas

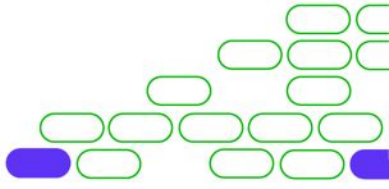
Demonstração

Faculdade XPe

Aula 5.1.1. – Bancos de dados em Nuvem

Nesta aula

- Características de bancos de dados em nuvem e seus desafios na área de processamento distribuído e gestão dos dados.



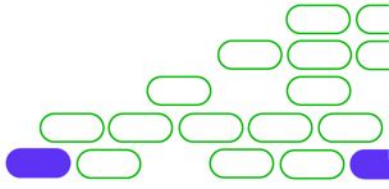
Introdução

- Computação em nuvem
  - Modelo computacional onde os recursos de software e hardware são disponibilizados por um fornecedor externo à empresa, chamado de nuvem.

- “Computação como serviço”
  - As necessidades de recursos computacionais, como hardware e software são atendidas como um serviço, sob demanda.

Introdução

- Computação em nuvem - Características
  - Elasticidade
  - Auto atendimento
  - Custos proporcionais ao uso, não mais à disponibilidade dos recursos.
  - Alta disponibilidade



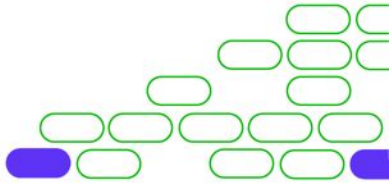
Introdução

- Computação em nuvem – Tipos de serviços
  - IaaS
    - Infraestrutura como serviço
  - PaaS
    - Plataforma como serviço
  - SaaS
    - Software como serviço
  - DBaaS
    - Banco de dados como serviço

Banco de Dados como Serviço

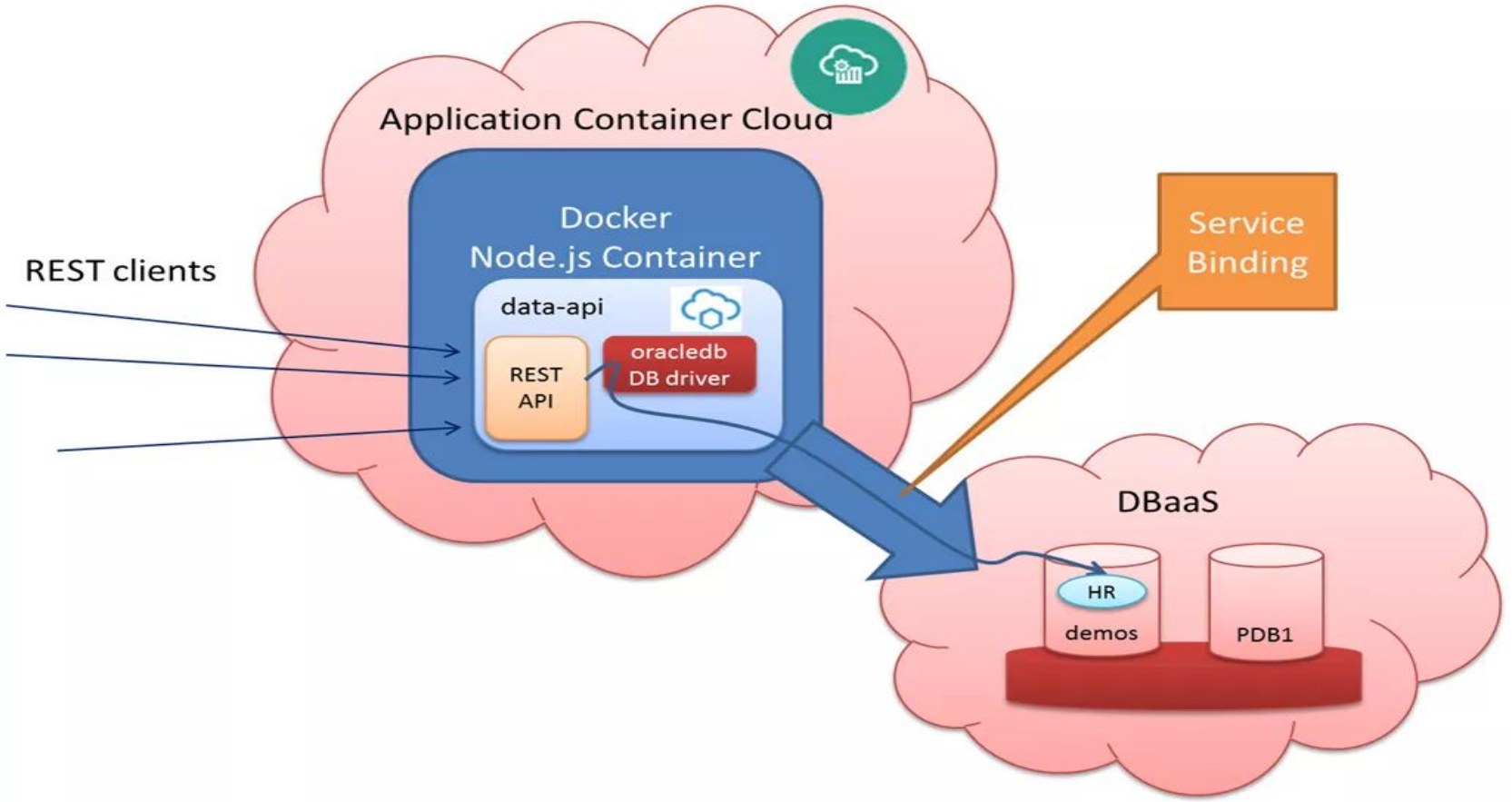
- Computação em nuvem, entre outros serviços, provê:
  - Bancos de dados como serviço (DBaaS)

- Definição:
  - Database as a Service (DBaaS) - Abordagem baseada em Cloud Computing para o armazenamento e gerenciamento de dados estruturados.



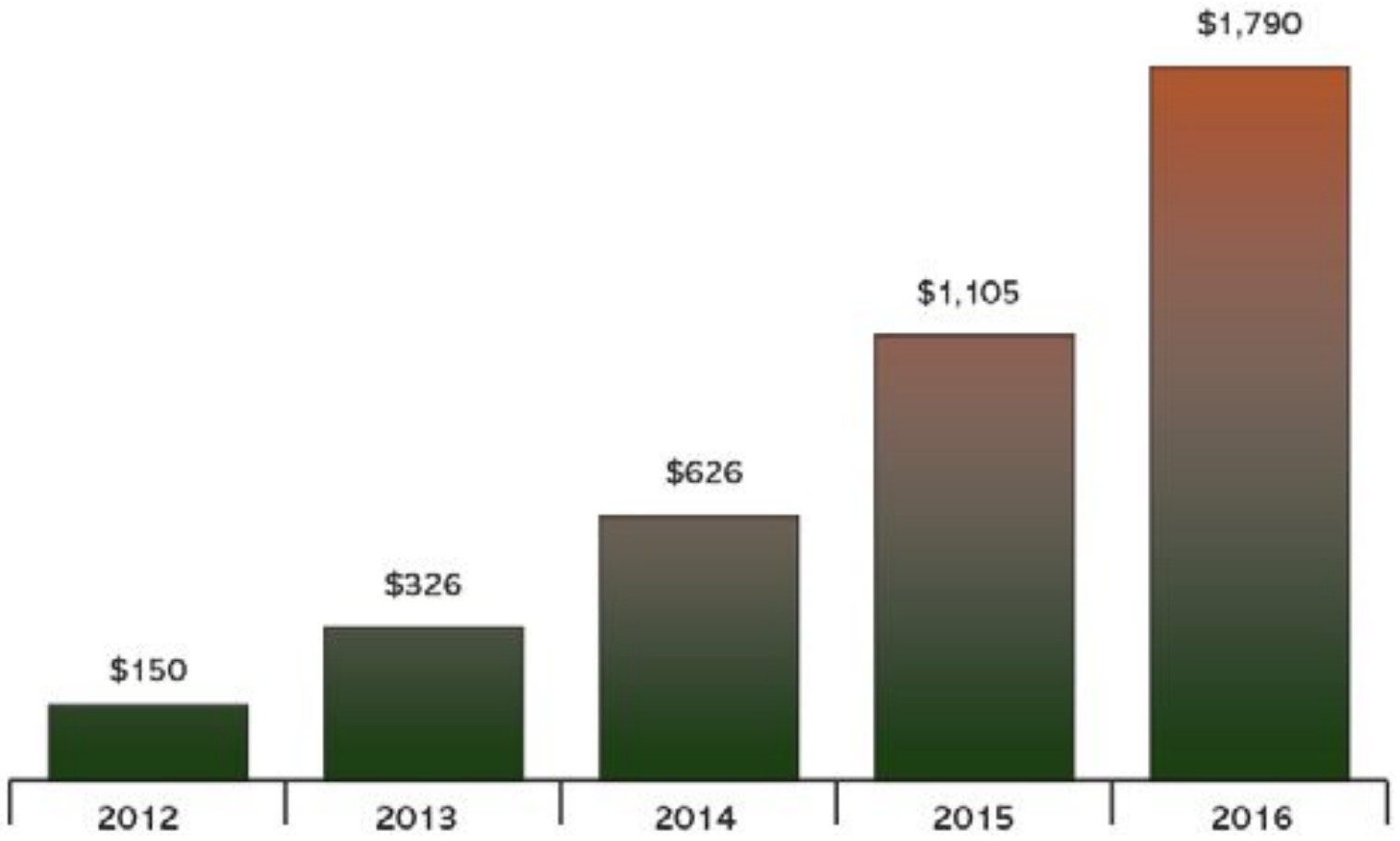
Banco de Dados como Serviço

- Bancos de dados como serviço (DBaaS)



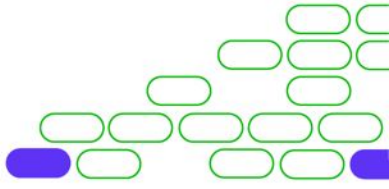
Introdução

- Crescimento do mercado de DBaaS (milhões de dólares)



Introdução

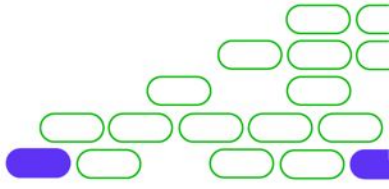
- Características
  - Infraestrutura de servidores terceirizada
  - Redução do tamanho das equipes
  - Alocação de recursos sob demanda (Elasticidade)
  - Automação de atividades de rotina
  - Integração com outros recursos disponíveis em nuvem



Preocupações

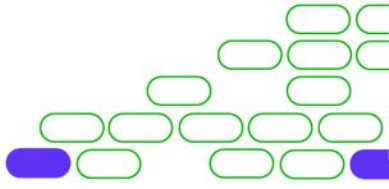
- Modo de armazenamento
  - Sistemas de arquivos
  - Redundância dentro das políticas de conformidade

- Acesso aos dados pelo fornecedor
- Localização física dos dados
- Dados confidenciais



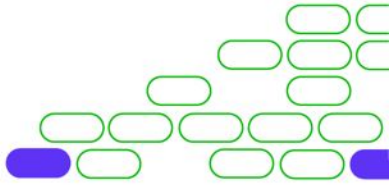
Novos desafios

- Disponibilidade
- Monitoramento
- Desempenho
- Confidencialidade
- Gestão de usuários
- Segurança de rede



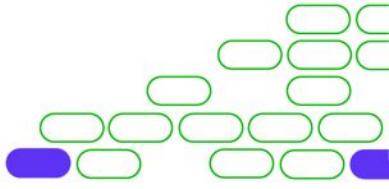
Novos desafios

- Disponibilidade
  - Uma das vantagens da computação em nuvem
  - Servidores são espelhados e distribuídos fisicamente em vários lugares
  - Caso um servidor fique indisponível os outros recebem as requisições
  - Em algumas situações, uma nova máquina é disponibilizada automaticamente



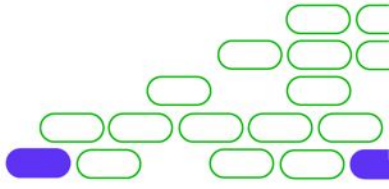
Novos desafios

- Monitoramento
  - Fornecedores de servidores em nuvem possuem ferramentas de monitoramento
  - O monitoramento fornece as seguintes informações:
    - Consumo de CPU
    - Consumo de Memória
    - Tráfego de rede
    - Usuários conectados



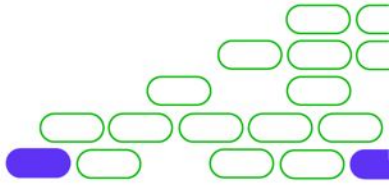
Novos desafios

- Desempenho
  - DBaaS permite que a carga seja balanceada por mais de um servidor
  - DBaaS permite que recursos computacionais sejam alocados dinamicamente, sob demanda



Novos desafios

- Confidencialidade
  - A gestão dos dados fica a cargo do fornecedor
    - Risco de acessos indevidos
  - Políticas de segurança adicionais se fazem necessárias
  - Risco de aplicações hospedadas na mesma nuvem terem acesso aos dados

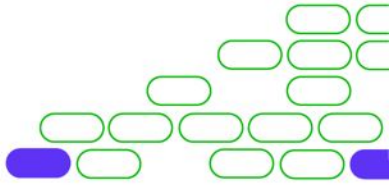


Novos desafios

- Gestão de usuários
  - Os usuários das aplicações e bancos de dados permanecem da mesma forma
  - Os administradores de dados, do cliente, devem ter atenção especial aos acessos administrativos
  - Políticas de auditoria devem ser consideradas com maior intensidade nesses ambientes

Novos desafios

- Segurança de rede
  - O acesso ao banco de dados passa pela internet, portanto um risco adicional
  - A latência de rede é um requisito que devem ser avaliado na implantação do acesso aos serviços
  - Se todas as camadas de serviço estiverem na nuvem, o problema de latência é mitigado
  - Conexões SSL podem contribuir para melhor segurança



Conclusão

- ☑ O crescimento de bancos de dados em nuvem mudam o paradigma de gestão dos dados.
- ☑ Os bancos de dados como serviço possuem vantagens que podem trazer competitividade às empresas.
- ☑ Equipes menores e mais capacitadas são necessárias.
- ☑ É importante pensar em políticas específicas para esse cenário.

Faculdade XPe

Aula 5.2.1. – Introdução aos Bancos de Dados Distribuídos

Nesta aula

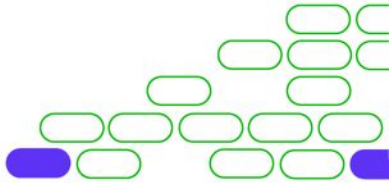
- Conceitos em bancos de dados distribuídos

Introdução

- A distribuição dos dados, processamento e armazenamento oferece algumas vantagens e desafios.

- Existem diferentes arquiteturas de bancos de dados distribuídos e nessa aula iremos discutí-las.

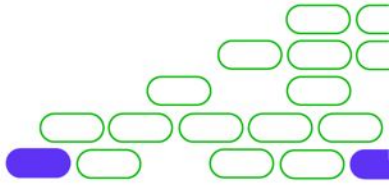
- Os principais fornecedores de SGBDs de mercado possuem soluções distribuídas.



Motivação

- Uma empresa multinacional possui sua operação integrada por uma base de dados única.

- Problemas:
  - Cada filial terá uma cópia dos dados?
  - Cada filial terá uma parte dos dados?
  - Como armazenar os dados comuns entre filiais?
  - Como minimizar o tráfego de dados entre as filiais?
  - Como manter a integridade referencial entre as filiais?



Motivação

- Porque, então, distribuir os dados?
  - Aumento da disponibilidade
  - Acesso distribuído – mais próximo do local onde é consultado
  - Facilidade em aumentar a estrutura e volume de dados

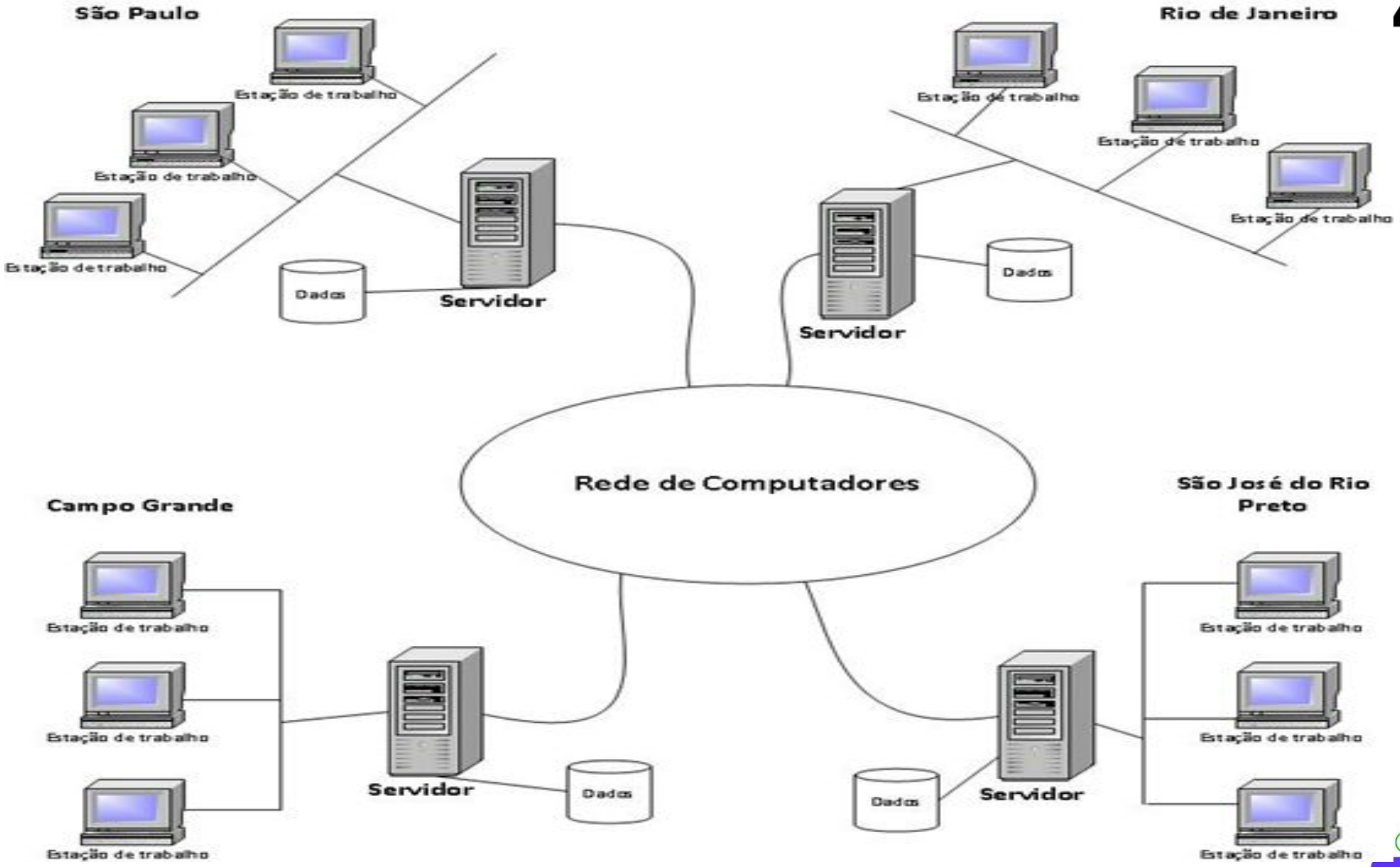
- Quais situações utilizar?
  - Empresas distribuídas geograficamente
  - Redes de hóteis
  - Empresas aéreas
  - Cadeias de produção integradas

Conceitos

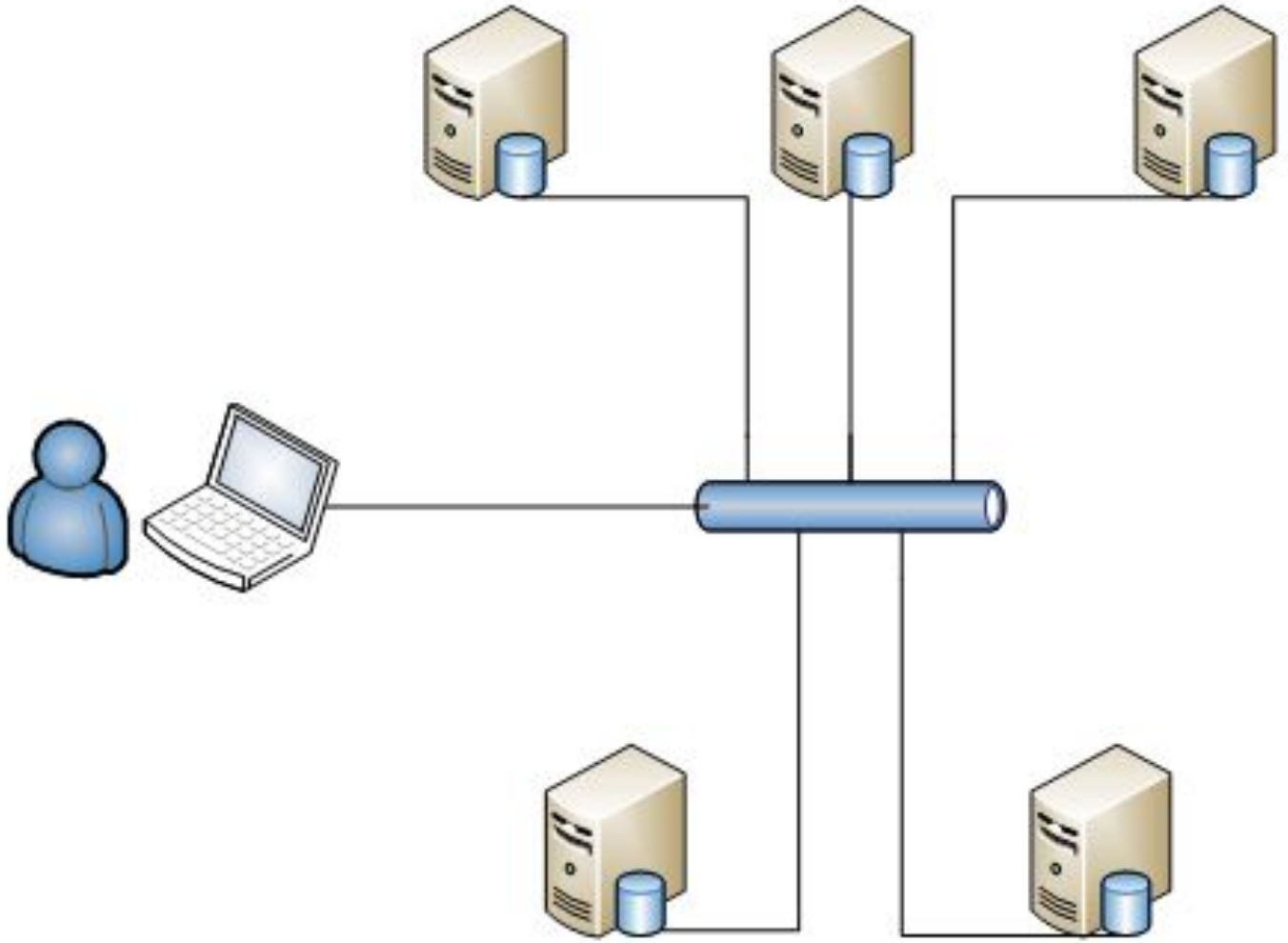
- Banco de Dados Distribuído
  - Vários bancos de dados locais, interligados através de uma rede.
  - Processamento prioritariamente local.

- Sistema Gerenciador de Bancos de Dados Distribuídos (SGBDD)
  - Sistema que gerencia um banco de dados distribuído
  - Mantém a visão centralizada e única para o usuário
  - Processamento distribuído.

BDD - Arquitetura

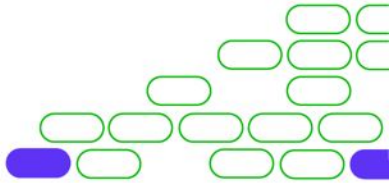


SGBDD - Arquitetura



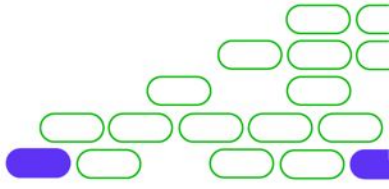
Desafios

- Controle de concorrência
  - Como manter integridade dos dados, mesmo com acessos concorrentes?
- Replicação
  - Como manter cópia dos dados, em cada ponto do banco de dados distribuído, sincronizados?
- Confiabilidade
  - Como manter as informações disponíveis?
  - Como garantir as propriedades ACID?



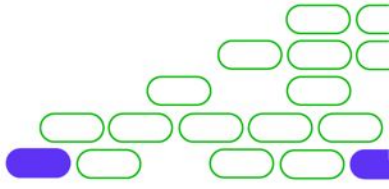
Controle de concorrência

- Controle de concorrência
  - Métodos para garantir que os dados são alterados na ordem cronológica correta.
  - Garantir que o banco de dados esteja consistente antes e depois de uma transação.
  - Garantir todas as propriedades de uma transação, para todas as transações do banco de dados.



Propriedades ACID – Relembrando...

- Propriedades ACID
  - Atomicidade
    - Transações são indivisíveis
  - Consistência
    - As restrições de integridade são obedecidas
  - Isolamento
    - Cada transação acontece sem interferência de outras transações. Como se fossem sequenciais.
  - Durabilidade
    - Todas as operações de uma transação são gravadas, de forma permanente, sem perda de informações.



Protocolos de controle

- Os SGBDs distribuídos possuem protocolos para implementação do controle das transações.

- Exemplos:
  - Two-phase-commit (2PC)
  - Locks (Bloqueios)
  - Two-phase-locks (2PL)

Two-phase-commit (2PC)

- Para que o SGBD não aguarde indefinidamente o commit dos dados, em razão de problemas de rede.

- O commit é dividido em dois momentos:
  - Fase 1: Preparação
    - A transação envia uma mensagem preliminar, para “avisar” que irá acontecer um commit.
    - Cada nó recebe a mensagem e responde se consegue consolidar essa alteração localmente.
  - Fase 2: Commit

Two-phase-commit (2PC)

- O commit é dividido em dois momentos:
  - Fase 2: Commit
    - Após a resposta de todos os nós do SGBD, o gerenciador de transações deve decidir se confirma as alterações.

    - Regras:
      - Se um nó solicita que a transação deve ser abortada, então a transação será abortada em todos os nós.
      - Se todos os nós confirmarem a confirmação da transação, então o commit acontece.

Locks

- Locks, ou Bloqueios, é um mecanismo mais comum em bancos de dados distribuídos.
  - A transação que vai modificar os dados, solicita o bloqueio desses dados ao nó gerenciador.
  - O nó gerenciador solicita aos demais nós, o bloqueio desses registros para outras transações.
  - O nó gerenciador controla os objetos bloqueados
  - Quando a transação recebe a confirmação do bloqueio, realiza as alterações.
  - O bloqueio é retirado quando o commit da transação acontece.

Conclusão

- ☑ Os bancos de dados distribuídos são soluções criadas no passado para manter bancos de dados globais.
- ☑ A partir dos conceitos principais, como separação dos dados e controle de transações, surgiram soluções comerciais.
- ☑ As soluções comerciais implementam parcialmente os conceitos de SGBDs distribuídos.
- ☑ O SGBD distribuído tem como vantagens o desempenho, disponibilidade e escalabilidade.

Faculdade XPe

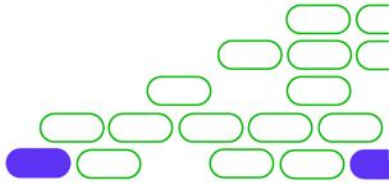
Aula 5.2.3. – SGBD – Google Bigtable

Nesta aula

- Banco de dados da Google: Bigtable

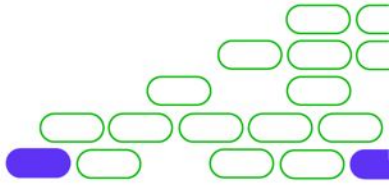
Introdução

- O Google Bigtable é o banco de dados projetado pela Google, para suportar:
  - Aplicações com grande volumes de dados semi-estruturados.
  - Resolver problemas de escala nas aplicações da Google.
  - Não existia nenhum SGBD capaz de suportar o volume de dados e transações.
  - Otimizado com o sistema de armazenamento da Google, o que provê desempenho.



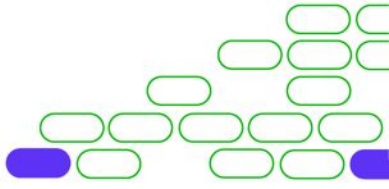
Características

- Grandes volumes de dados
- Milhões de máquinas
- Aplicações heterogêneas
- Milhões de usuários simuntâneos
- Dados semi-estruturados
- Otimização em armazenamento de baixo nível
  - Sem camada de aplicação (SGBD)
- Suporte a operações MapReduce



Características

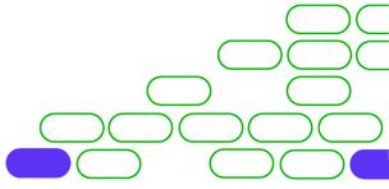
- SGBD tolerante a falhas
- Modelo de dados orientado a colunas
- Escalabilidade
  - Milhares de servidores
  - Terabytes de dados em memória
  - Petabytes de dados em disco
  - Milhões de operações de leitura e escrita
- Ajustes dinâmicos
- Compactação de dados em disco



Componentes

- Chubby
  - Realiza o controle de concorrência
  - Implementa política de Locks
  - Mantém a permissão de acessos

- Google File System
  - Armazenamento dos dados fisicamente
  - Balanceamento de carga em nível de disco

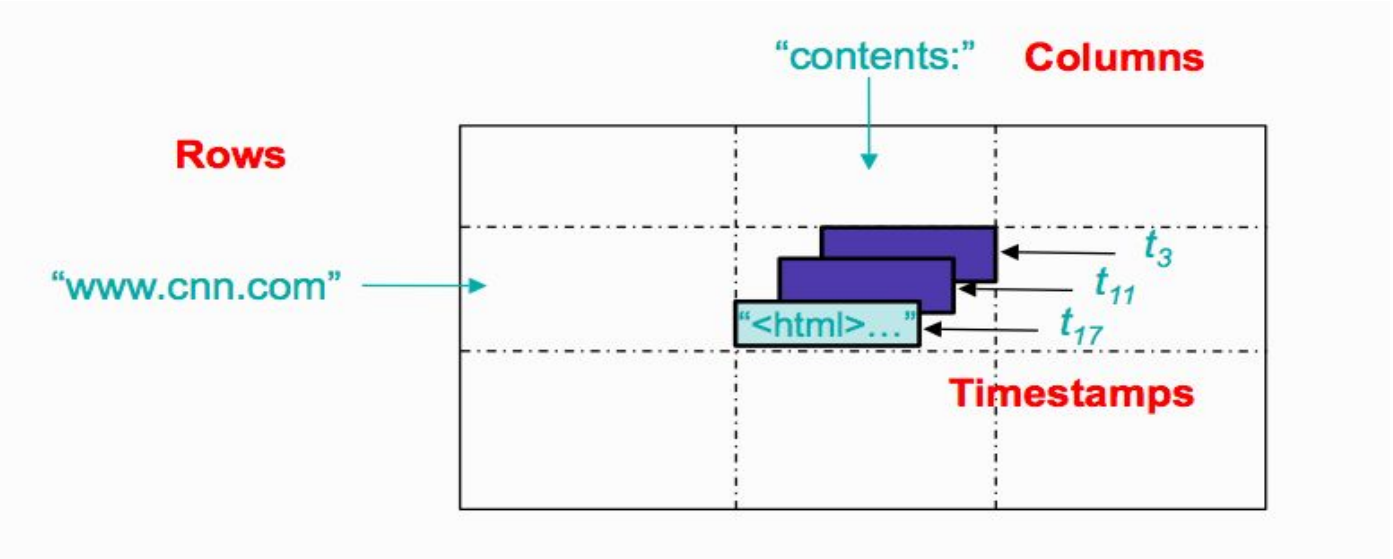


Aplicações

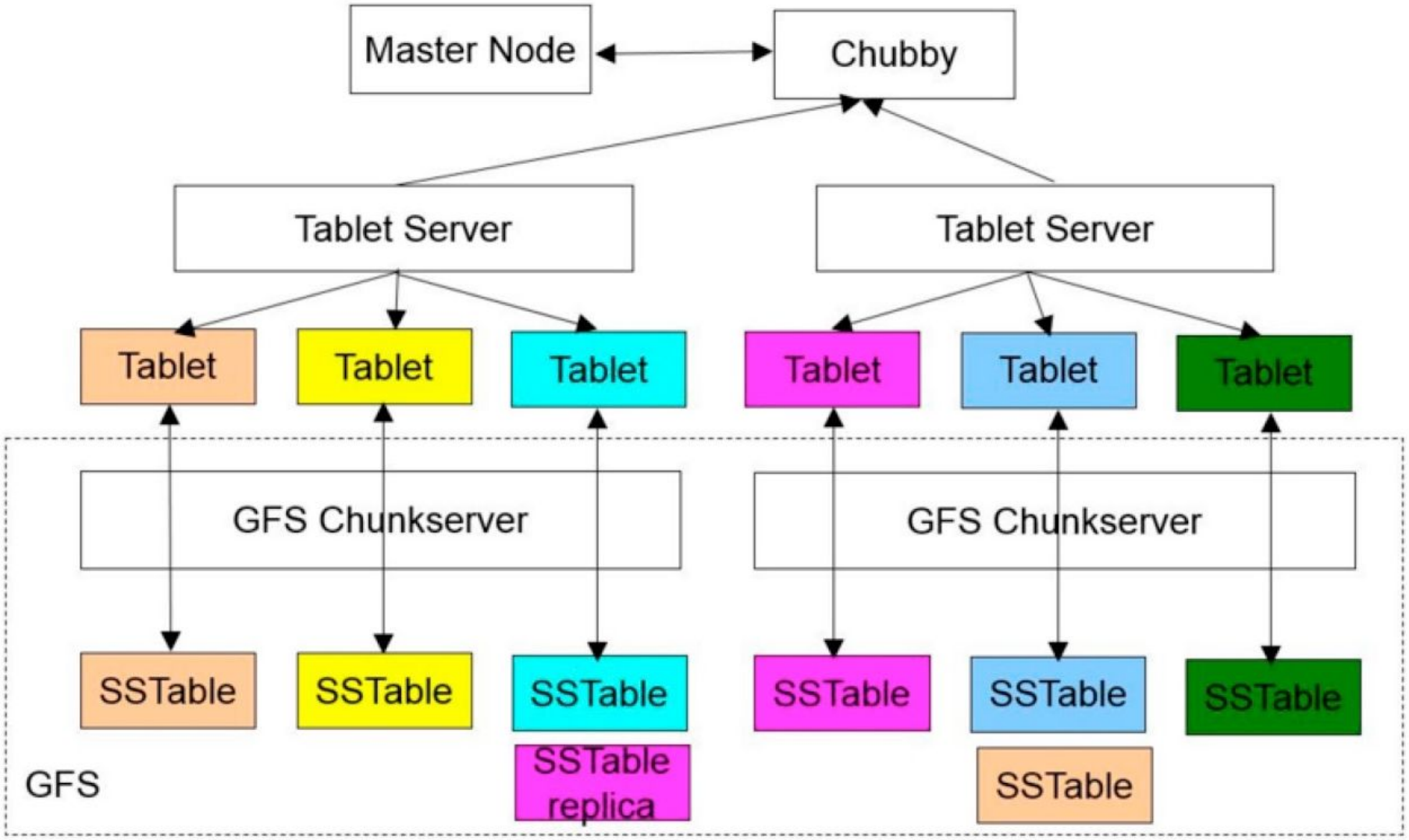
Project name	Table size (TB)	Compression ratio	# Cells (billions)	# Column Families	# Locality Groups	% in memory	Latency-sensitive?
Crawl	800	11%	1000	16	8	0%	No
Crawl	50	33%	200	2	2	0%	No
Google Analytics	20	29%	10	1	1	0%	Yes
Google Analytics	200	14%	80	1	1	0%	Yes
Google Base	2	31%	10	29	3	15%	Yes
Google Earth	0.5	64%	8	7	2	33%	Yes
Google Earth	70	–	9	8	3	0%	No
Orkut	9	–	0.9	8	5	1%	Yes
Personalized Search	4	47%	6	93	11	5%	Yes

Modelo de Dados

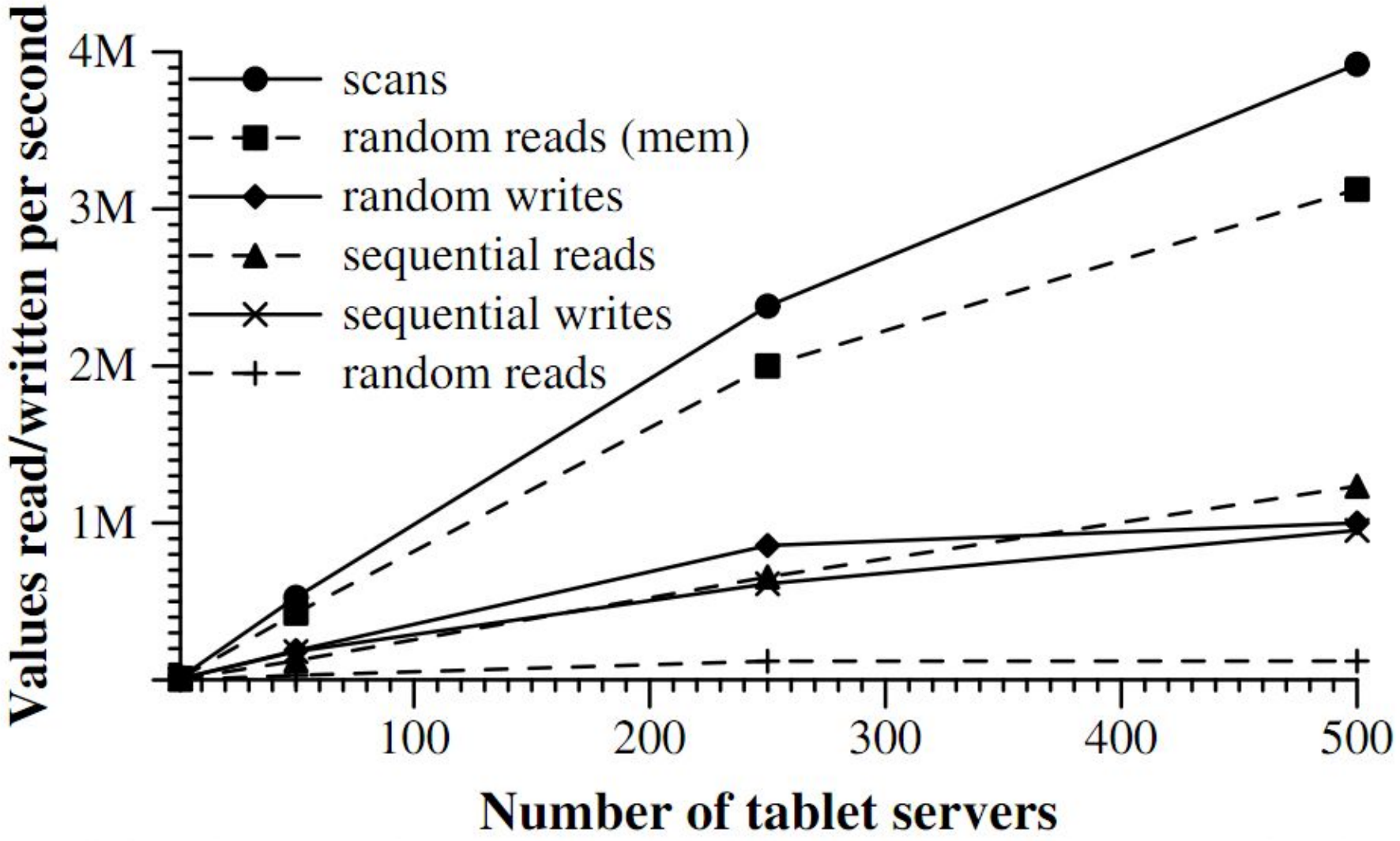
- Família de colunas
  - Mapa multi-dimensional ordenado.
- Indexado pela linha, coluna e timestamp.
- Cada valor no mapa é um array de bytes não interpretado.



Arquitetura

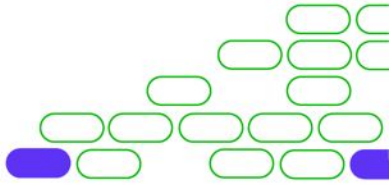


Desempenho


Values read/written per second
4M
3M
2M
1M
scans
random reads (mem)
random writes
sequential reads
sequential writes
random reads
100
200
300
400
500
Number of tablet servers

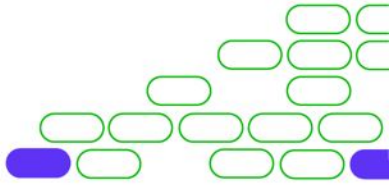
Desempenho

- Fatores que favorecem o desempenho
  - Balanceamento de carga
  - Algoritmo de rebalanceamento que reduz movimento de registros
  - Aumento dinâmico de tablet servers de acordo com a utilização
  - A maior parte das leituras é realizada e memória



Conclusão

- ☑ Bigtable é um SGBD que atendeu os objetivos de suportar as aplicações da google.
- ☑ Sua arquitetura distribuída, escalável e com integração com o Google File System provê desempenho às aplicações.
- ☑ O modelo de dados não-relacional flexibiliza as alterações nos dados, reduzindo número de bloqueios.
- ☑ Esse banco de dados foi base para a criação de outros no paradigma NoSQL, como exemplo o Apache Cassandra.

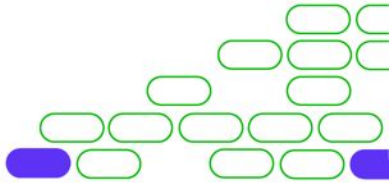


Faculdade XPe

Aula 5.2.4. – SGBD Distribuído Oracle Rac

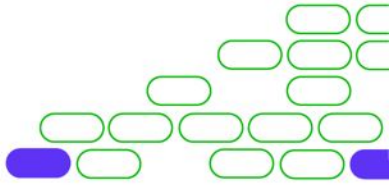
Nesta aula

- Solução da Oracle para Bancos de Dados distribuídos

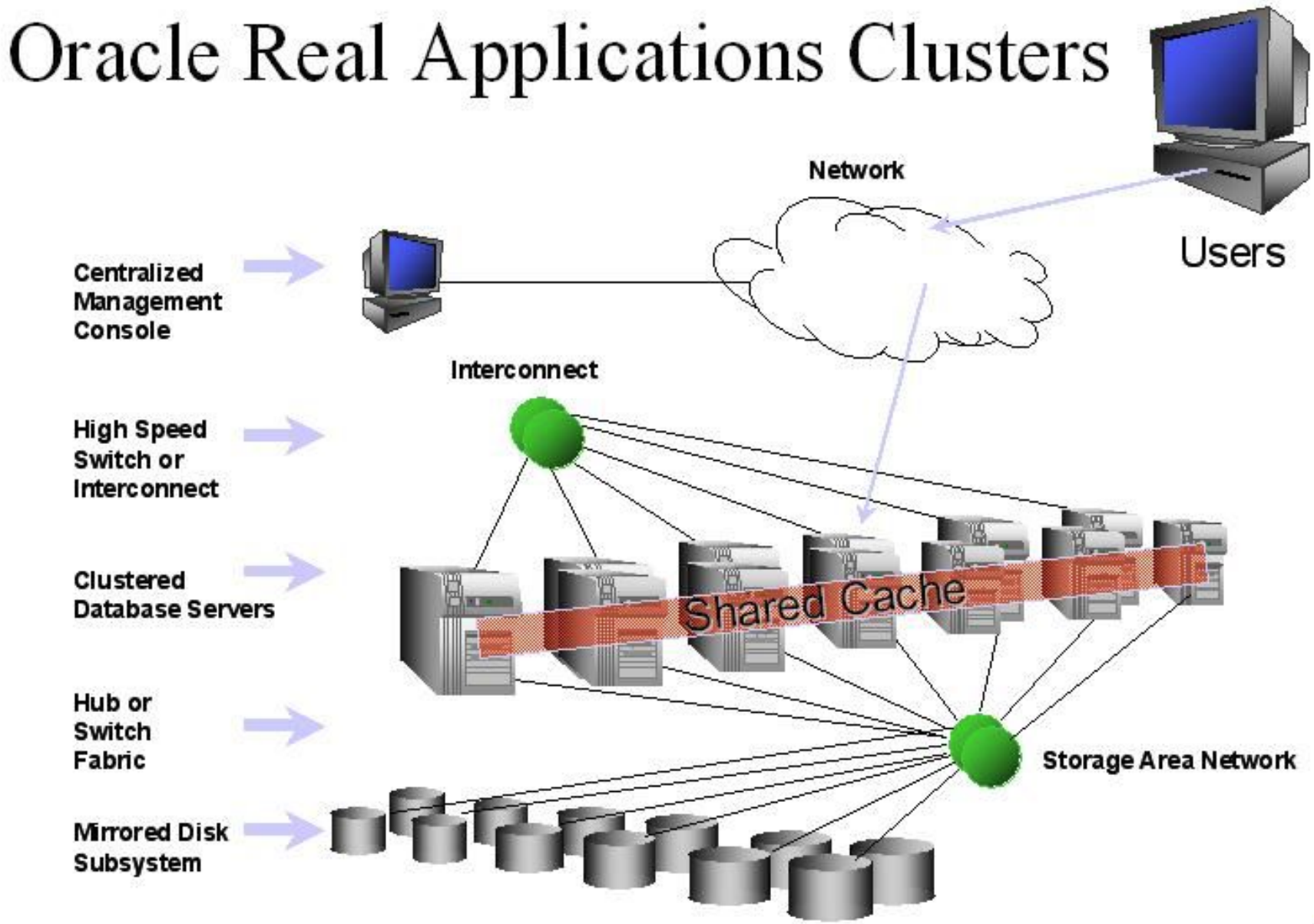


Introdução

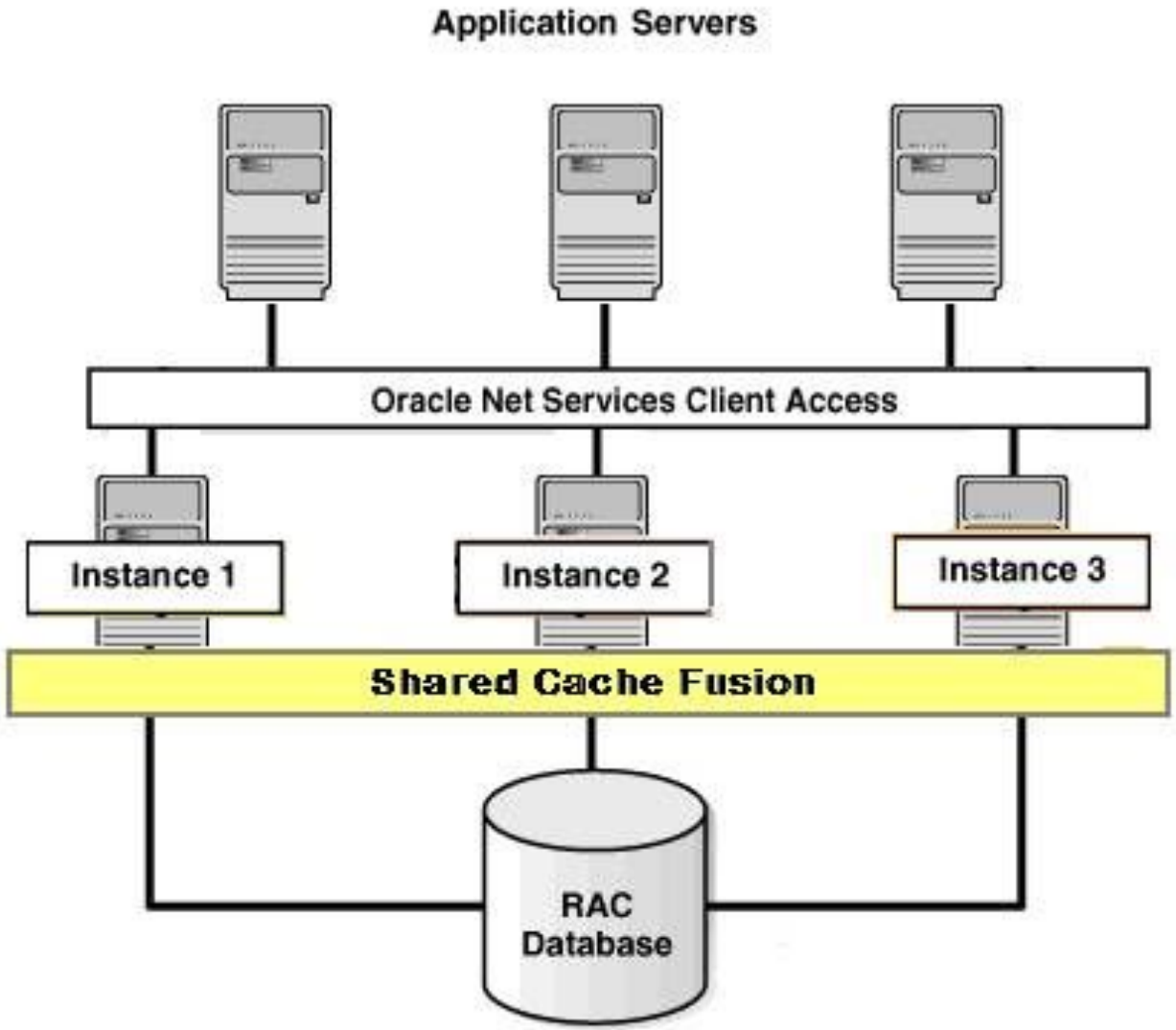
- A partir da versão 9i da Oracle, a demanda de usuários cresceu substancialmente, era necessária uma arquitetura que suportasse volumes maiores.
- Desafios
  - Disponibilizar um ambiente redimensionável
  - Redução de custos
  - Eliminação de ponto único de falha
- Oracle Real Application Clusters
  - SGBD Oracle que funciona em um cluster



Oracle RAC - Arquitetura



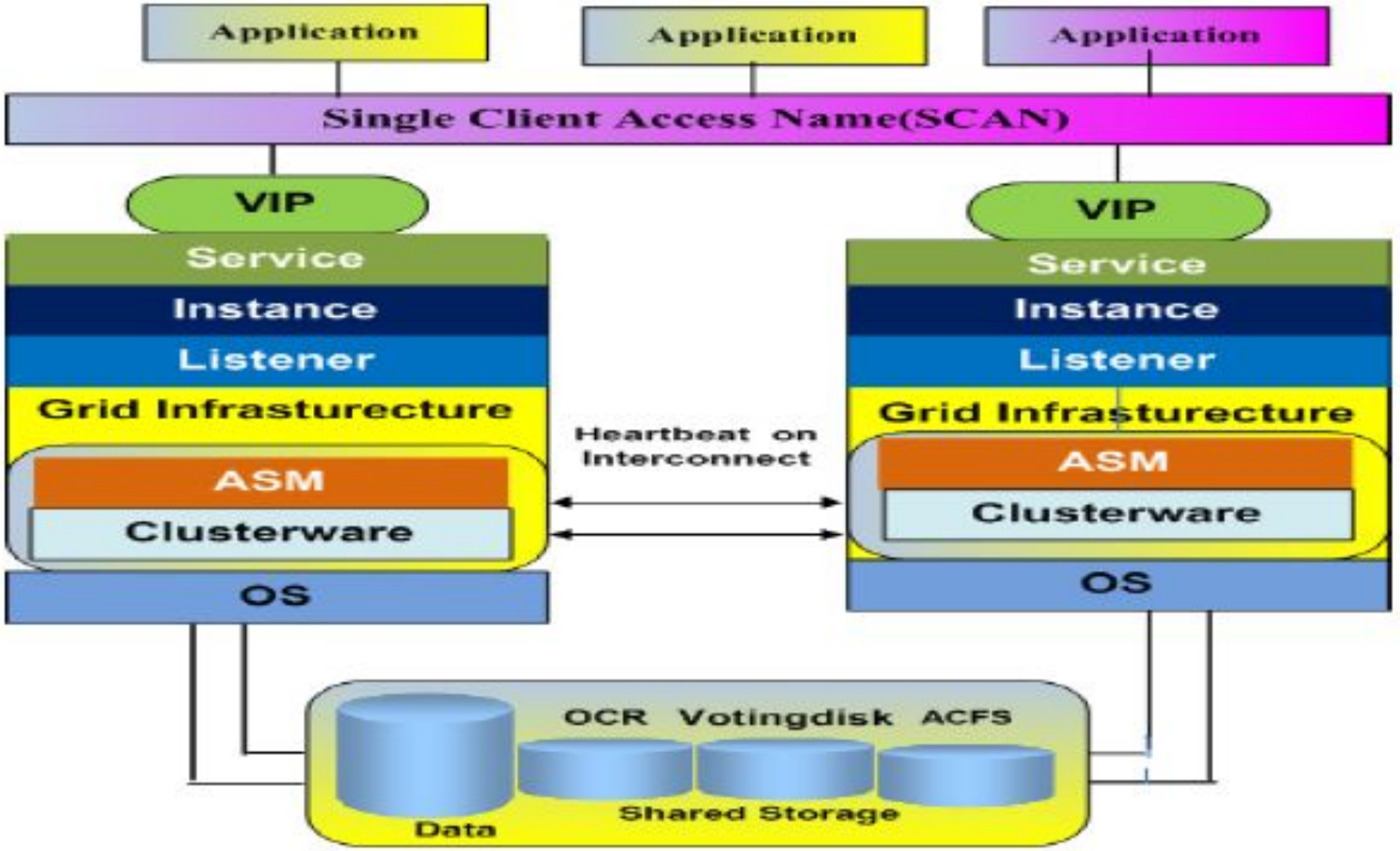
Oracle RAC – Arquitetura Básica



Componentes

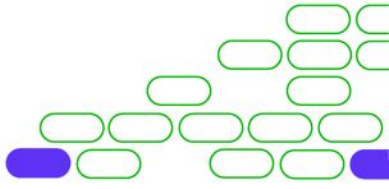
- Oracle Clusterware
  - Componente desenvolvido para realizar a integração de forma transparente e uniforme entre os servidores
  - Esse componente é um pré-requisito para a instalação do Oracle RAC.
  - Monitoramento dos bancos de dados do RAC.
    - Quando um nó no cluster é iniciado, todas as instâncias, listeners e serviços são iniciados automaticamente.
    - Se ocorre uma falha em uma instância, o clusterware reinicia automaticamente a instância, de forma transparente.

Introdução

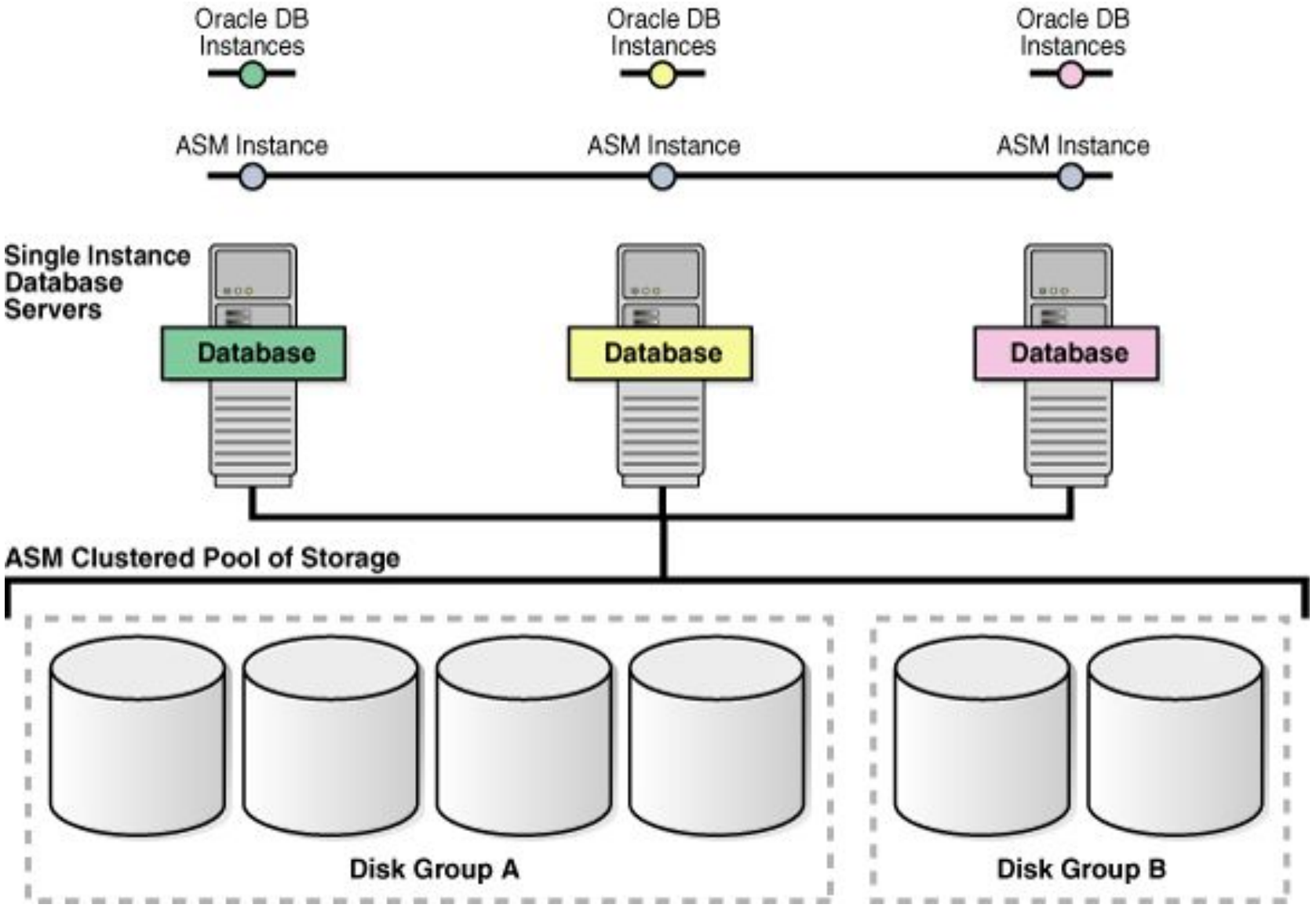

Application
Application
Application
Single Client Access Name(SCAN)
VIP
VIP
Service
Instance
Listener
Grid Infrastructure
ASM
Clusterware
OS
Heartbeat on Interconnect
Service
Instance
Listener
Grid Infrastructure
ASM
Clusterware
OS
OCR
Votingdisk
ACFS
Data
Shared Storage

Componentes

- Automatic Storage Management (ASM)
  - Sistema de arquivos com gerenciamento automático.
  - O ASM possui alto desempenho em leitura e escrita de forma assíncrona.
  - Possui recurso de distribuição de carga próprio.
  - Independente de sistema operacional



Componentes



Componentes

- Endereço VIP
  - O RAC exige um endereço IP virtual para cada servidor no cluster.
  - O endereço IP virtual é um endereço IP não usado na mesma subrede que a rede local.
  - Este endereço é usado por aplicações para se conectarem ao banco de dados RAC.
    - Se ocorre uma falha em um nó, é realizado um failover no IP virtual para outro nó no cluster
  - Recurso que minimiza o tempo de recuperação, pois não necessita aguardar a rede local.

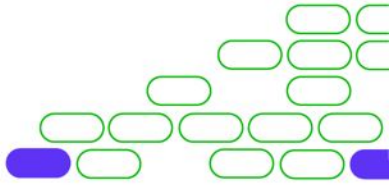
Componentes

- Interconnect
  - Funções Heartbeat / Keep Alive.
  - Troca de mensagens.
  - Troca de informações sobre locks e deadlocks.
  - Atualização de cache (Cache Fusion).

- ADR – Automatic Diagnostic Repository
  - Ferramenta para detecção de problemas
  - Centraliza os logs de todos os componentes em um local

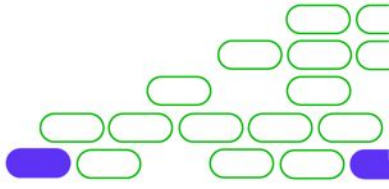
Benefícios

- Alta disponibilidade
  - Confiabilidade
  - Capacidade de recuperação
  - Detecção de erro
  - Operação contínua
- Escalabilidade
- Ferramentas de gestão especializadas
- Balanceamento de carga

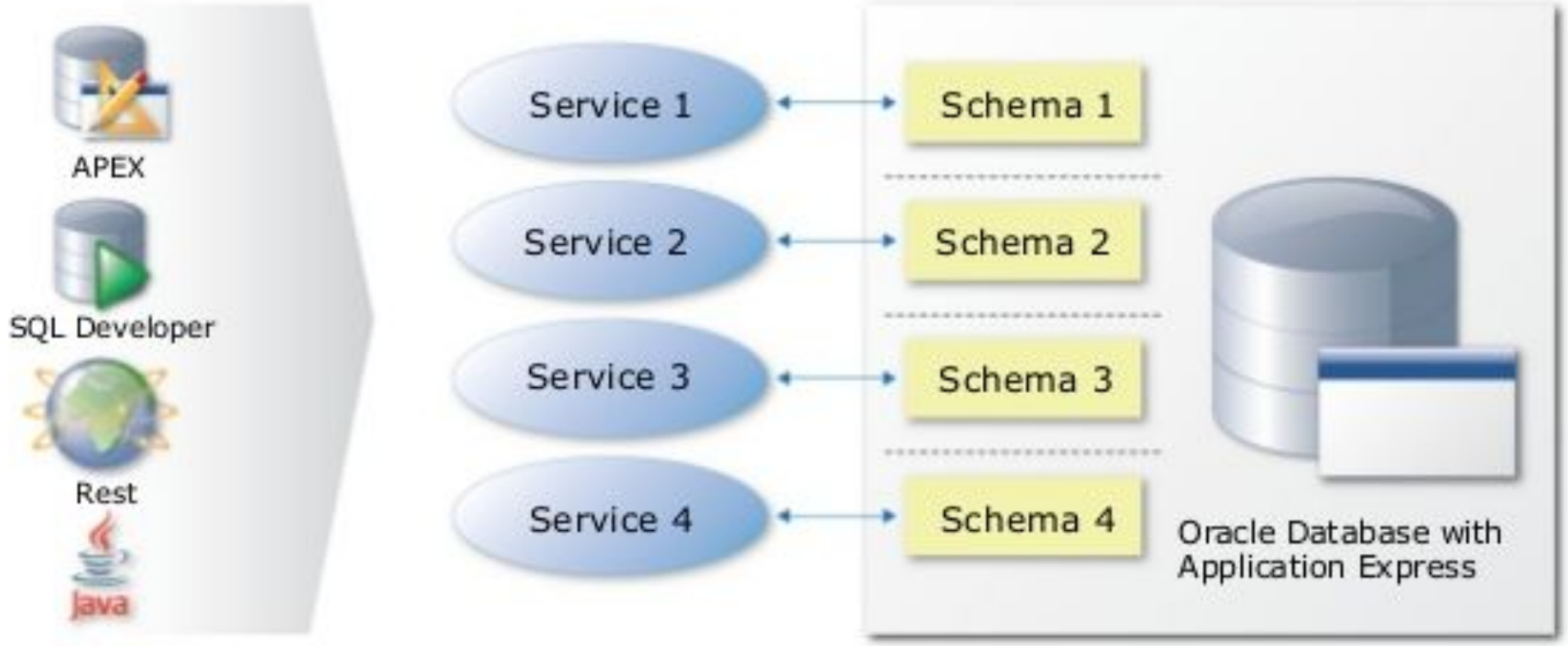


Database Cloud Service

- As versões mais recentes do SGBD Oracle existem, também, como serviço.
- O Oracle Database Cloud Service pode ser oferecido com todas as features do SGBD tradicional.
- Escalabilidade
  - Permite inclusão de memória e disco dinamicamente.
  - Possui a versão Oracle RAC em nuvem.
  - Possui componentes de alta disponibilidade, além da disponibilidade da nuvem.



Solução em nuvem



One Service = One Schema = One Tablespace = One Data File

Conclusão

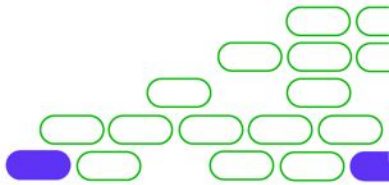
- ☑ O SGBD Oracle desenvolveu solução em arquitetura de processamento distribuído.
- ☑ Fornece mecanismos de gerenciamento e manutenção do cluster de maneira simplificada.
- ☑ As soluções atualmente existem em nuvem e são integradas com outras soluções oracle, ex: Golden Gate.
- ☑ Possui as vantagens de ser um SGBD consolidado e ter ferramentas de auto gestão eficientes.

Faculdade XPe

Aula 5.3.1. – Sistemas de Arquivos Distribuídos

Nesta aula

- Sistemas de armazenamento de arquivos distribuídos



Introdução

- O serviço de armazenamento de arquivos, em sistema distribuído, traz os seguintes benefícios:
  - Alta disponibilidade
  - Escalabilidade
  - Tolerância a falhas
  - Redundância

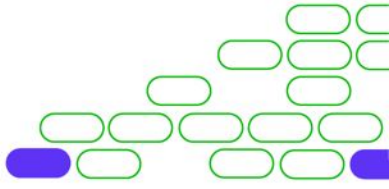
- Além de utilização de hardware e software com baixo custo.

Características

- Segurança
  - Permissão de acesso aos arquivos
  - Hierarquia de acesso
  - Identidade dos usuários
- Transmissão dos dados
  - Criptografia das informações

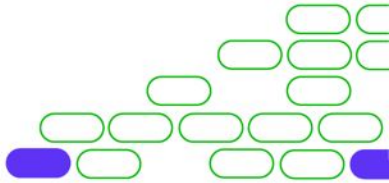
Características

- Consistência
  - Cache em clientes (menor sobrecarga no servidor)
  - Validação as versões de arquivos no cliente
- Travamento
  - Acesso exclusivo nos arquivos por parte do cliente
  - Clientes podem liberar e renovar o travamento dos arquivos
- Replicação
  - As cópias de um arquivo são atualizadas após as alterações



Tipos de Sistemas

- DAS - Direct-attached storage
- NAS - Network Attached Storage
- SAN - Storage Area Network



Direct-attached storage

- DAS – Armazenamento Diretamente conectado
  - Dispositivos de armazenamento conectados diretamente nos computadores.
  - Aumenta a capacidade de armazenamento local
  - Facilita o compartilhamento das informações
  - Gerenciamento de dados mais complexo

NAS - Network Attached Storage

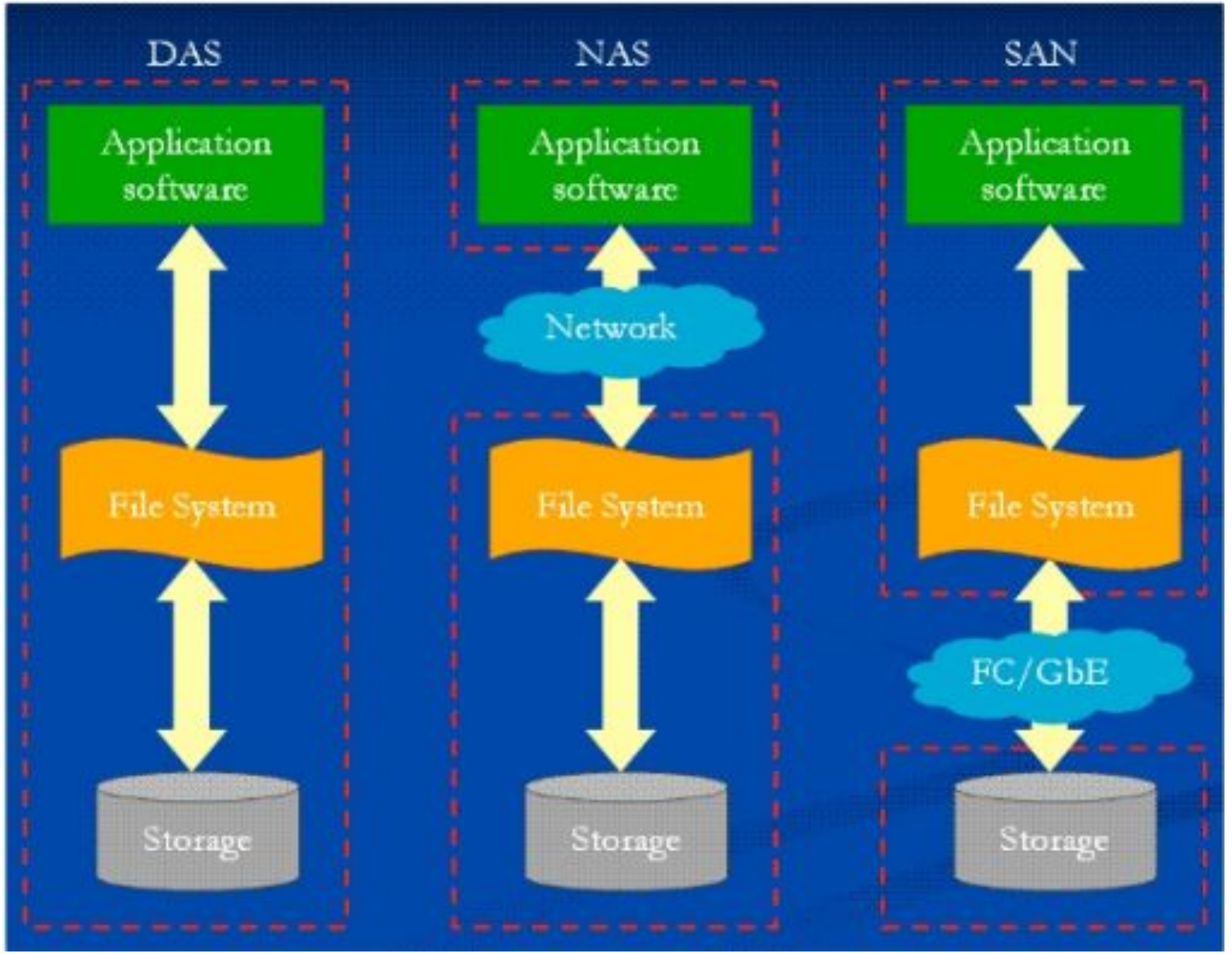
- NAS – Armazenamento conectado por rede
  - Modelo utilizado em maior quantidade atualmente
  - Utiliza protocolos próprios para acesso e comunicação
  - Existem softwares especializados nesse tipo de aplicação
  - Transparente para o cliente

SAN - Storage Area Network

- SAN – Rede de Armazenamento
  - Rede composta por servidores e storages.
  - Rede isolada da rede local

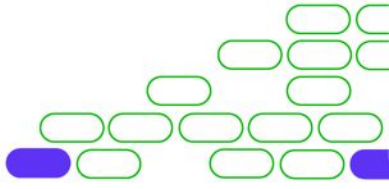
- Rede com implementação das seguintes tecnologias:
  - Fibre Channel
  - Ethernet

Comparativo



Aplicação - NFS

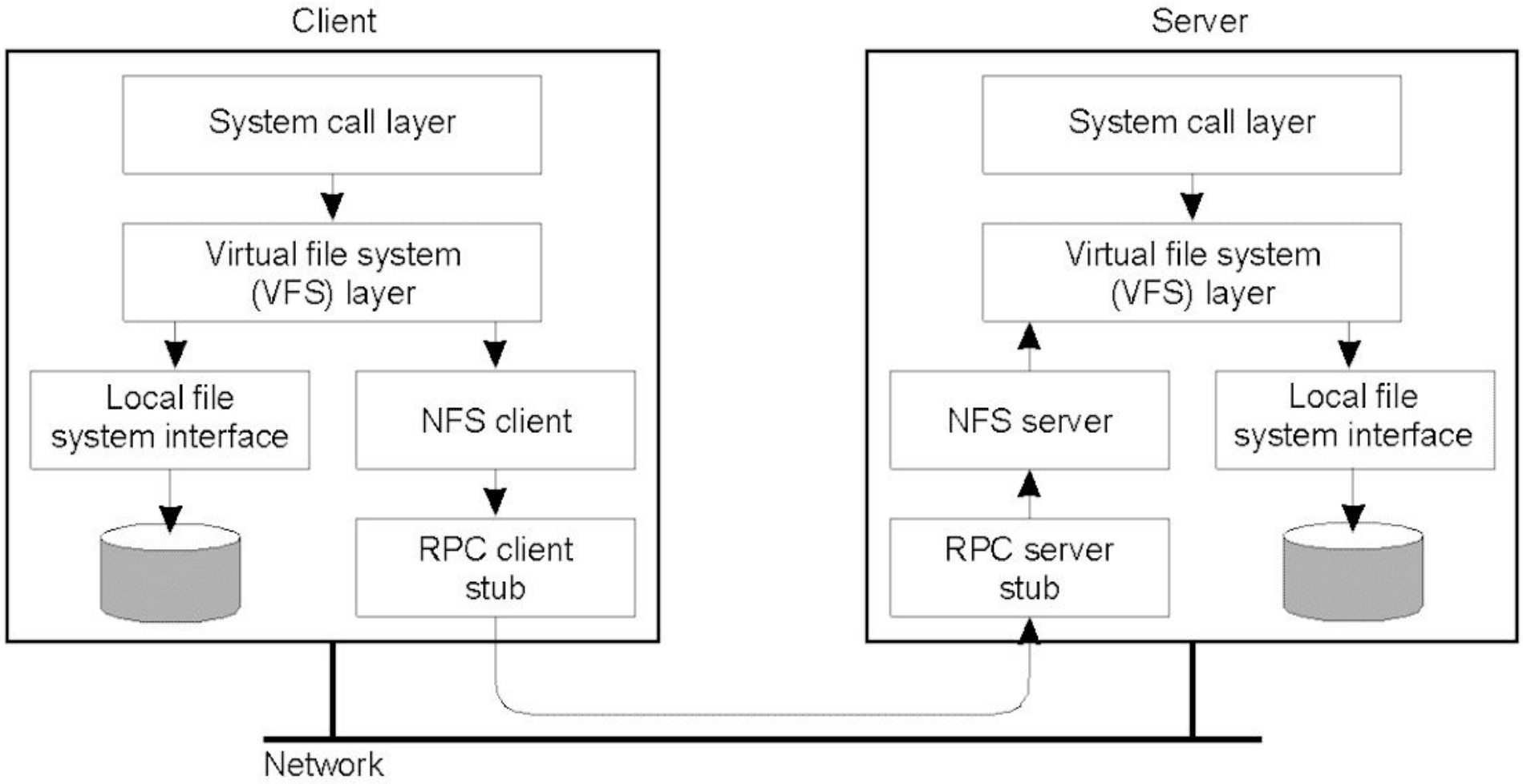
- Network File System
  - Sistema de arquivos distribuído, que permite o acesso ao usuário remotamente
  - Usuário manipula os arquivos como se fossem locais
  - Possui comandos para criação, leitura e remoção dos arquivos
  - Criado pela Sun, em 1985, primeiro sistema de arquivos comercializado como uma solução



Aplicação - NFS

- Clientes e servidores na mesma rede local
- Transparência de acesso
- Transparência de localização
- Independência de sistema operacional

Aplicação - NFS



Conclusão

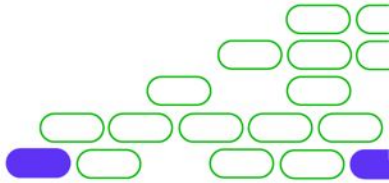
- ☑ Os sistemas de arquivos distribuídos são poderosas ferramentas para gestão dos arquivos.
- ☑ Permitem aumento de capacidade computacional de forma transparente
- ☑ Permitem alta disponibilidade e desempenho
- ☑ Demanda cuidados maiores em questões de segurança
- ☑ Dependem do bom funcionamento das redes de computadores relacionadas

Faculdade XPe

Aula 5.3.2 – Google File System

Nesta aula

- Características do sistema de arquivos da Google, o Google File System (GFS)

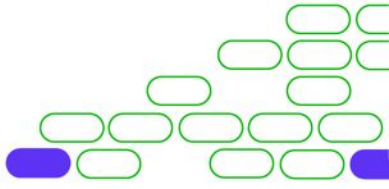


Introdução

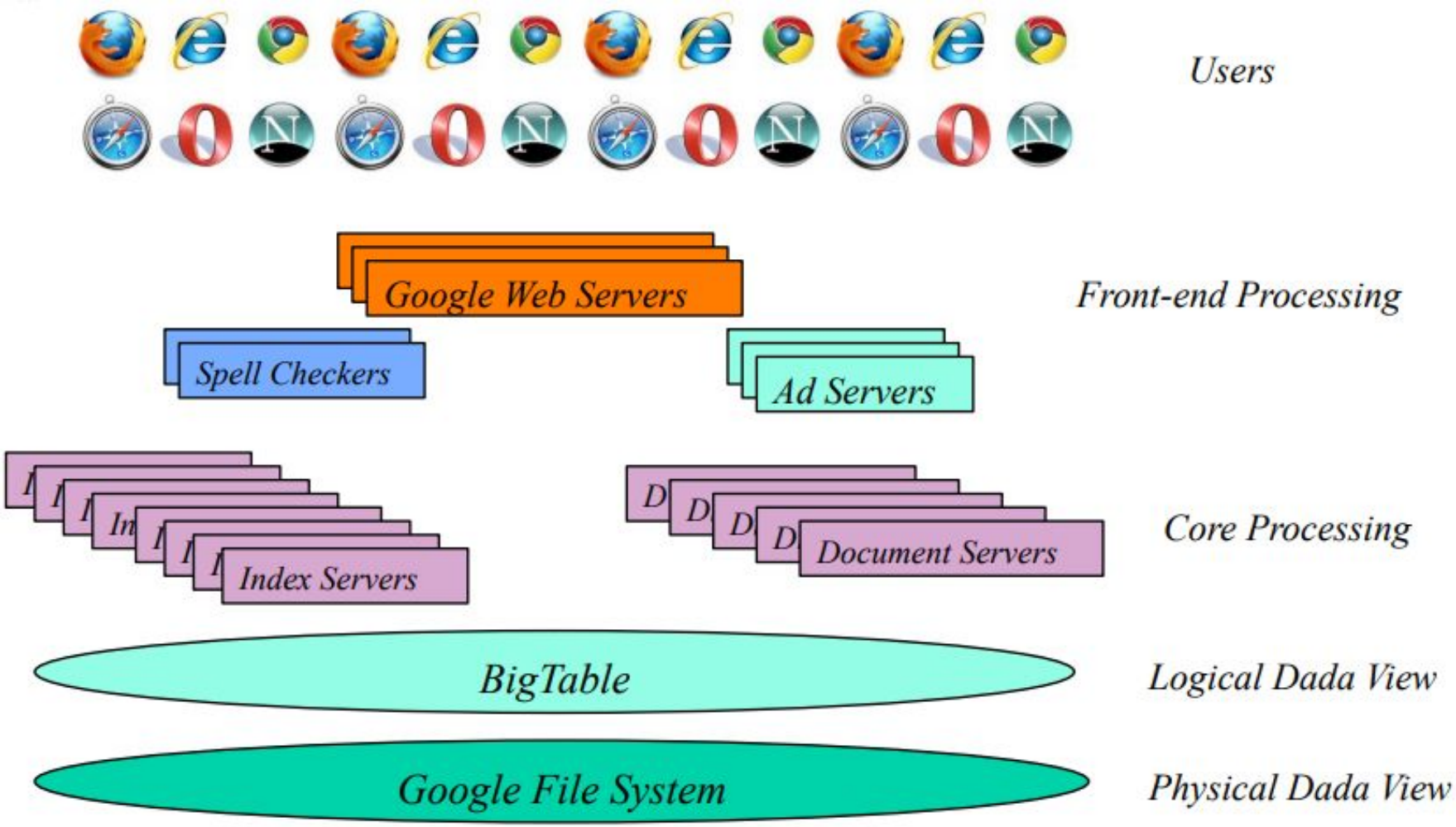
- A Google, em razão do enorme volume de dados que possui, desenvolveu um sistema de arquivos próprio para atender às demandas de suas aplicações.

- O Google File System foi construído com o objetivo de prover desempenho no armazenamento e recuperação de suas informações, e é um software privado.

- O GFS é parte da arquitetura das soluções de serviços da Google.

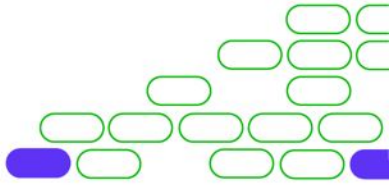


Arquitetura Google



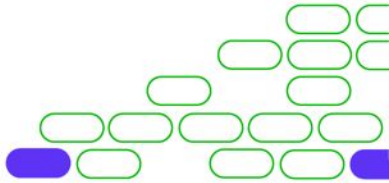
GFS - Características

- Considerando o perfil de suas aplicações, a Google definiu as seguintes características para seu sistema de arquivos:
  - Tolerância a falhas
  - Recuperação automática
  - Foco em arquivos grandes (A maioria em GB)
  - Crescimento rápido do conjunto de dados
  - Anexações recorrentes (record append)
  - Leitura sequencial de arquivos

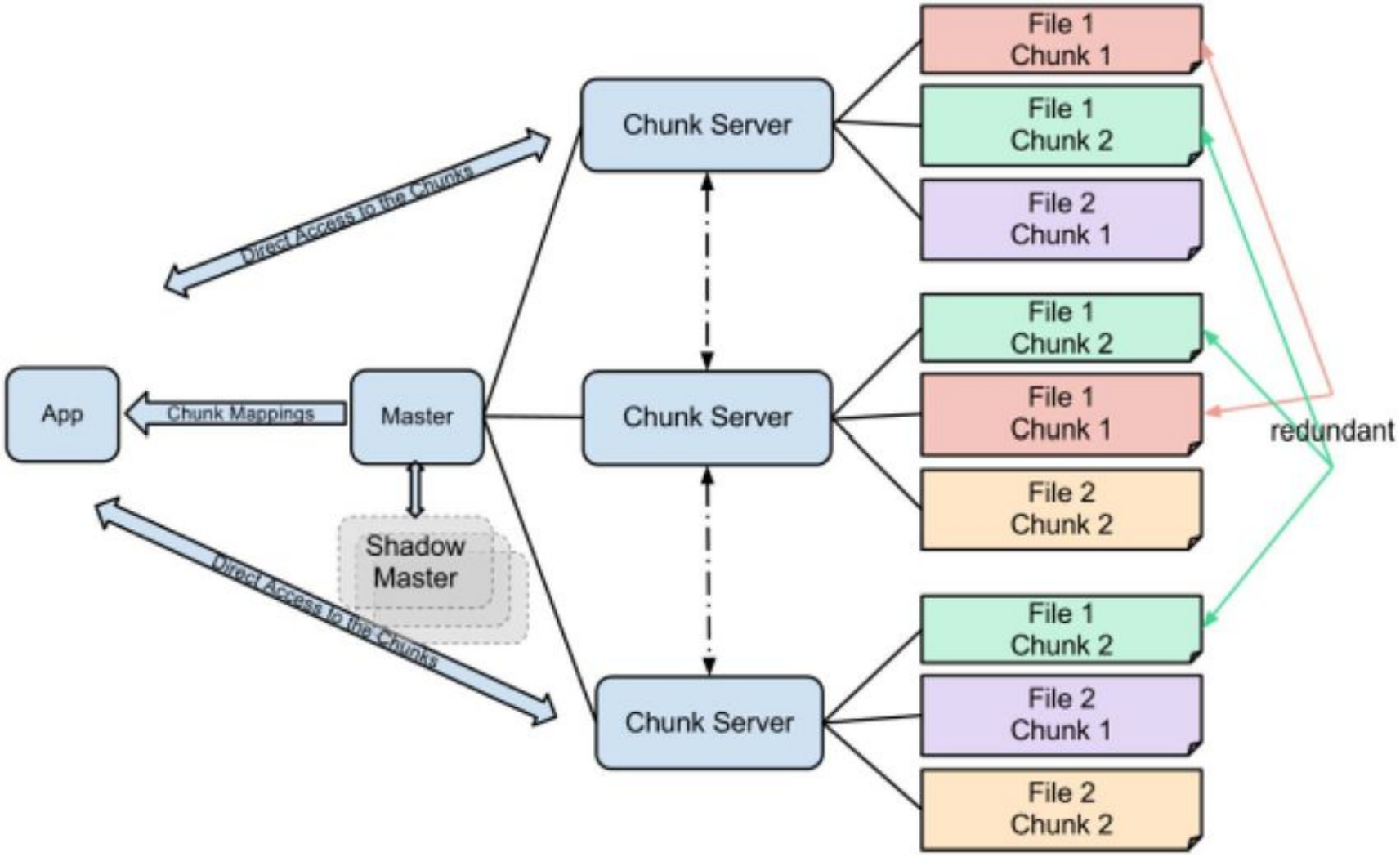


GFS – Aplicações Suportadas

- Algumas aplicações que utilizam o GFS
  - Youtube
  - Google Earth
  - Blogger
  - Gmail
  - Google Maps
  - Google Sugest
  - Etc.

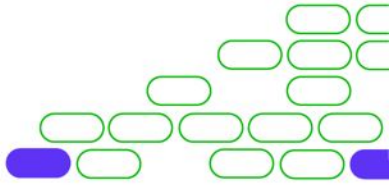


GFS - Arquitetura

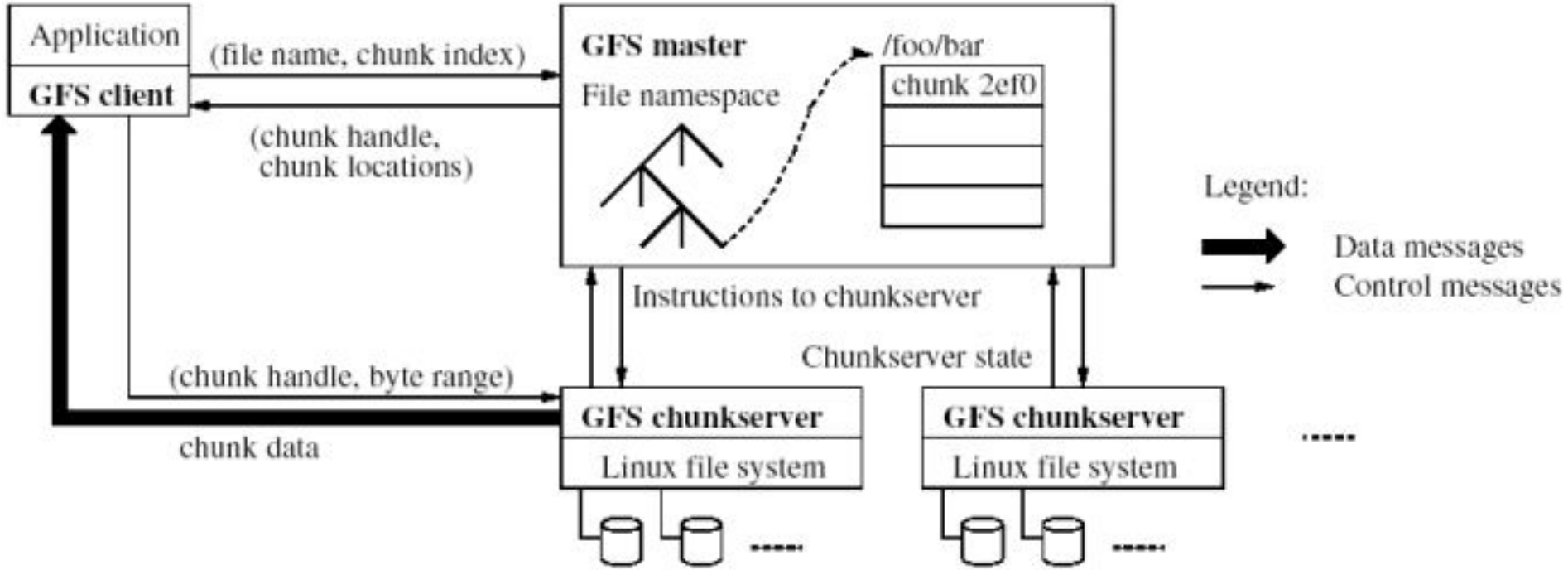


GFS - Componentes

- Master Server
  - Responsável pela coordenação do cluster
- Chunkserver
  - Componente responsável pelo armazenamento dos arquivos
- Chunk
  - Componentes similares a blocos de dados.
  - Um arquivo pode conter um ou mais chunks.



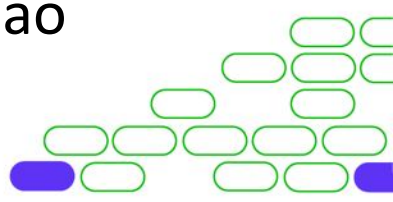
GFS - Funcionamento



- O cluster GFS possui um master e múltiplos chunkservers.
- Cada chunkserver pode ser acessado por vários clientes.
- Cada chunkserver possui vários chunks.

GFS - Operações

- Leitura
  - Aplicação informa (file name, byte range) para o Cliente GFS.
  - Cliente GFS traduz o pedido da aplicação para (file name, chunk index).
  - Master responde para o Cliente GFS com (chunk handle, replica locations).
- Gravação
  - Aplicação faz um pedido de escrita.
  - O Cliente GFS traduz o pedido (file name, data) para (file name, chunk index) e envia ao master.
  - O mestre responde com o (chunk handle, primary and secondary replica locations), localizações das réplicas.
  - O cliente envia dados de escrita para todas as localidades. Os dados são armazenados em buffers internos dos chunkservers

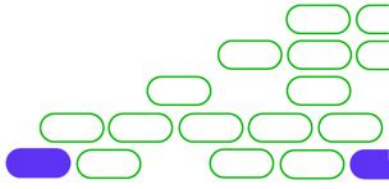


GFS - Operações

- Gravação (continuação)
  - O cliente manda o comando de escrita a réplica primária.
  - Esta réplica determina a ordem em série das instâncias dos dados armazenados no buffer e escreve nesta ordem no chunk.
  - A primária manda a ordem aos secundários, requisitando a escrita.
  - As secundárias respondem à primária.
  - A primária responde ao cliente.

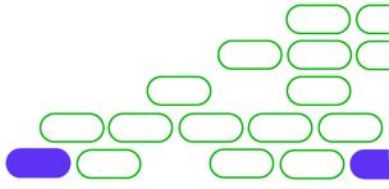
GFS - Operações

- Snapshot
  - Cria cópias de um arquivo ou uma árvore de diretório com baixo custo, ou seja, quase que em tempo real para minimizar interrupções de mudanças.
- Record Append
  - Permite que vários clientes gravem no mesmo local ao mesmo tempo garantido atomicidade
- Garbage Collector
  - No GFS quando a exclusão física não é feita imediatamente depois da exclusão lógica. Periodicamente, o garbage collector faz uma varredura nos níveis de chunks afim de procurar dados já excluídos



GFS - Operações

- Operação de Log
  - Os logs são utilizados no auxílio às operações de recuperação
  - Os logs registram o histórico de todas as alterações nos metadados e operações.
  - Os logs definem a ordem cronológica das operações
  - Existem operações de checkpoint, que fazem a replicação dos arquivos de log para manter o sincronismo nas diferentes réplicas dos dados



Conclusão

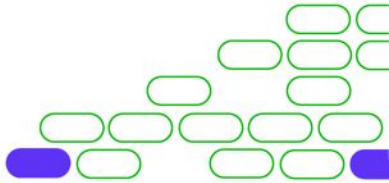
- ☑ O Google File System é um sistema de arquivos que processa dados em larga escala.
- ☑ Utiliza milhares de máquinas, em cluster, para o armazenamento das informações.
- ☑ Otimizado para as operações específicas, como anexação.
- ☑ Considera falha um evento normal, portanto tem mecanismos eficientes de tratamento e recuperação.

Faculdade XPe

Aula 5.3.3 – HDFS – Hadoop Distributed File System

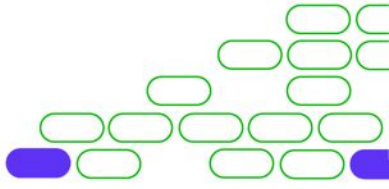
Nesta aula

- Sistemas de arquivos HDFS – Hadoop Distributed File System



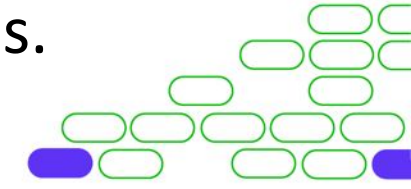
Introdução

- O Hadoop Distributed File System (HDFS) é um sistema de arquivos distribuído.
- Sistema de arquivos utilizado pela API do Hadoop
- O HDFS foi projetado com os seguintes requisitos, incialmente:
  - Funcionar de maneira distribuída, em hardware de baixo custo
  - Ser tolerante a falhas
  - Possuir bom desempenho na transferência de dados
  - Considerar grandes volumes de dados



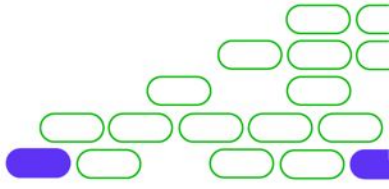
Características

- Falhas de Software
  - Em HDFS as falhas são tratadas como regra, não exceção.
  - Possui sistema de recuperação automática
- Streaming de dados
  - O HDFS foi projetado para processamentos em lotes
  - Taxas de transferência altas em transferências
- Grandes arquivos
  - A maioria dos arquivos possui gigabytes ou terabytes de dados.



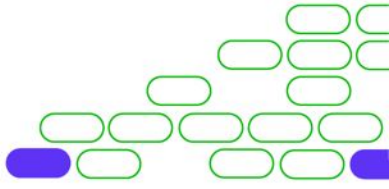
Características

- Modelo Simples
  - Simplicidade na manipulação de arquivos, com poucas operações.
- Transmissão de algoritmo, não de dados
  - Ao invés de transferir dados para serem processados, a arquitetura permite que o código seja transmitido.
- Portabilidade
  - Facilmente transportável entre plataformas heterogênas

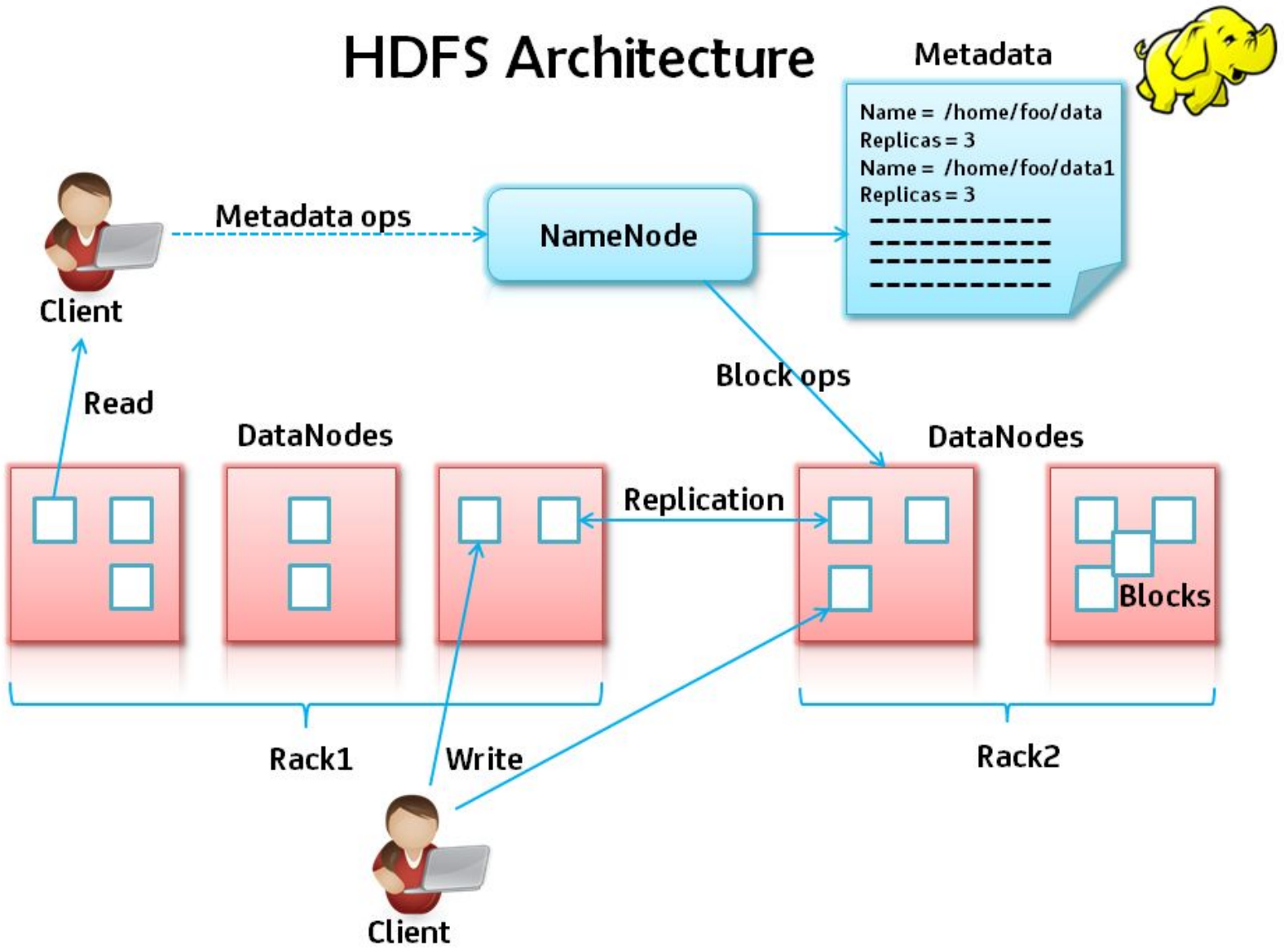


Componentes

- NameNode
  - Servidor gerenciador do sistema de arquivos
  - Define o controle de acesso
  - Armazena os metadados do sistema de arquivos
- DataNode
  - Conjuntos de blocos de dados, geralmente um por nó do cluster
- Namespace
  - Nome do sistema de arquivos que é exposto aos clientes

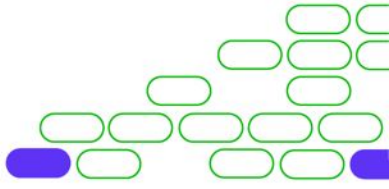


Arquitetura



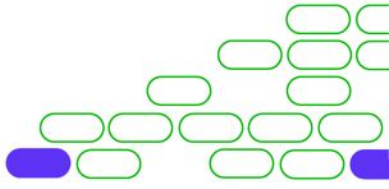
Replicação

- O NameNode periodicamente realiza a replicação dos dados.
- A replicação é operação importante para manter a tolerância a falhas.
- Réplicas de dados é armazenada em racks distintos



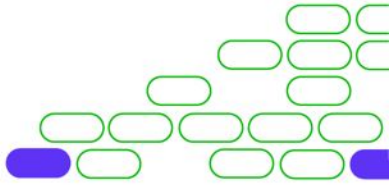
Funcionalidades

- Permissões e autenticação
- Rack Awareness
- Modo de segurança
- Balanceador
- Atualização e rollback de dados
- Nós de checkpoint
- Nós de backup

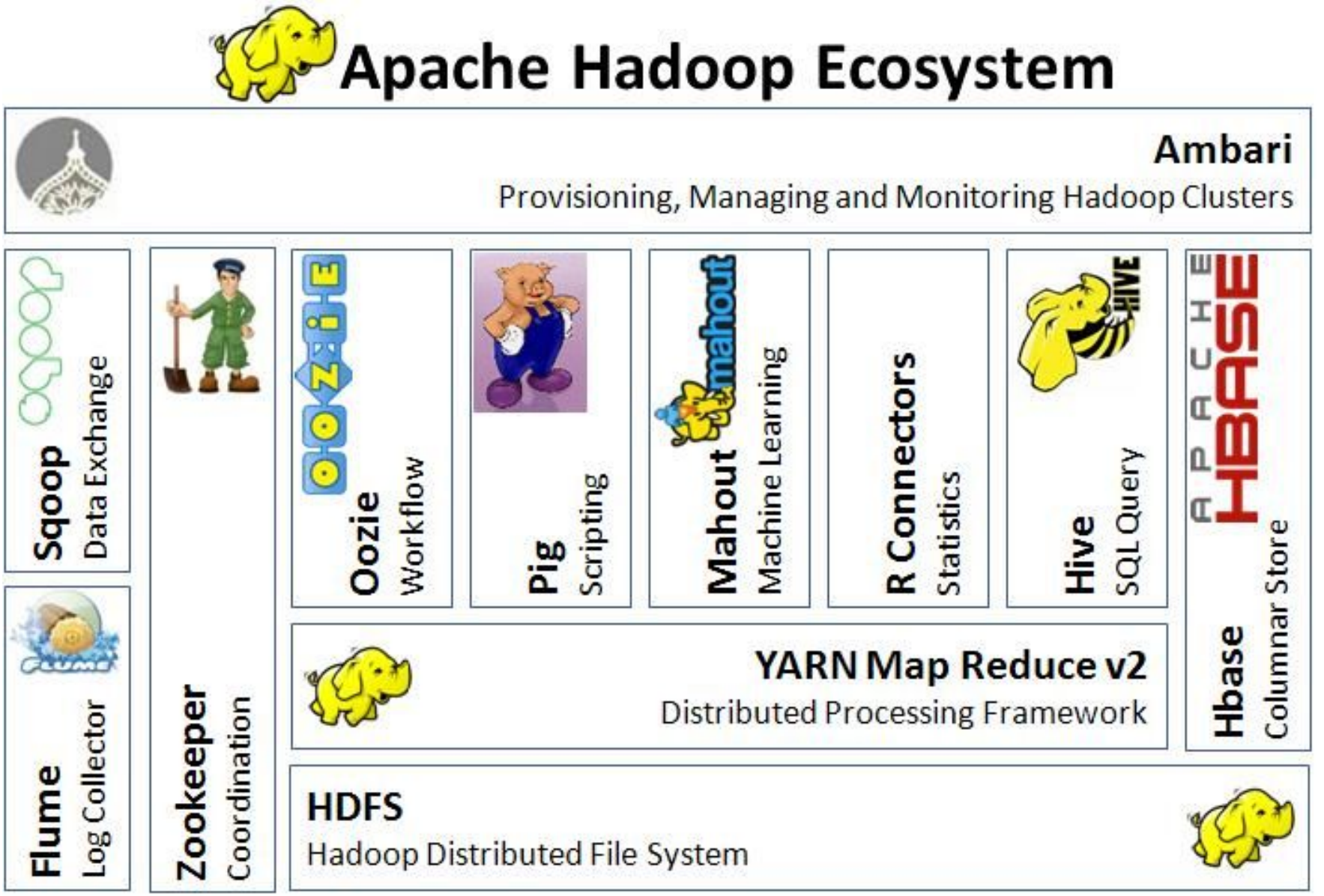


Alguns comandos

- O HDFS segue o padrão POSIX para seus comandos.
- Semelhantes aos comandos do linux.
- Exemplos:
  - cd, ls, mkdir, rm, rmdir

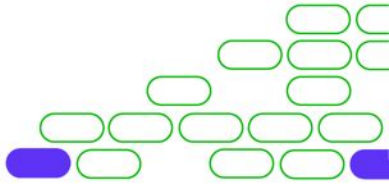


Ecossistema Hadoop



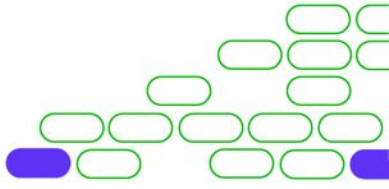
Aplicações que utilizam HDFS

- Facebook
- Adobe
- EBay
- Google
- IBM
- ImageShack
- Last.fm
- LinkedIn



Conclusão

- ☑ O HDFS é o sistema de arquivo distribuídos do hadoop framework.
  - ☑ O HDFS é altamente tolerante a falhas
  - ☑ É implementado para executar em clusters, com máquinas de baixo custo.
  - ☑ Segue o padrão do google file system
  - ☑ Realiza streaming dos dados com alto desempenho
  - ☑ Suporta grande parte das aplicações do ecossistema hadoop



Faculdade XPe

Aula 5.3.4 – Amazon S3

Nesta aula

- Amazon S3 - Sistemas de armazenamento de arquivos na web

Introdução

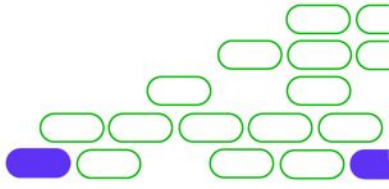
- O Amazon Simple Storage Service, conhecido como Amazon S3, é um armazenamento de objetos baseado em web service.

- O objetivo do projeto foi prover um serviço de armazenamento em nuvem.

- O Amazon S3 é um produto do portfolio de soluções em nuvem da Amazon

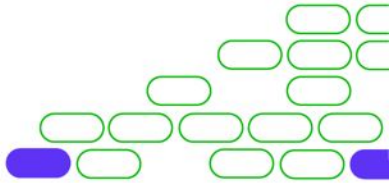
Características

- Serviço de armazenamentos de arquivos escalável
- Tem suporte a versionamento de arquivos
- Os objetos podem ser acessados via HTTP/HTTPS
- Os objetos podem ser acessados com protocolos de webservices, tais como REST e SOAP
- Executa replicação dos arquivos para manter durabilidade.
- Identifica dados corrompidos e faz o reparo dos erros



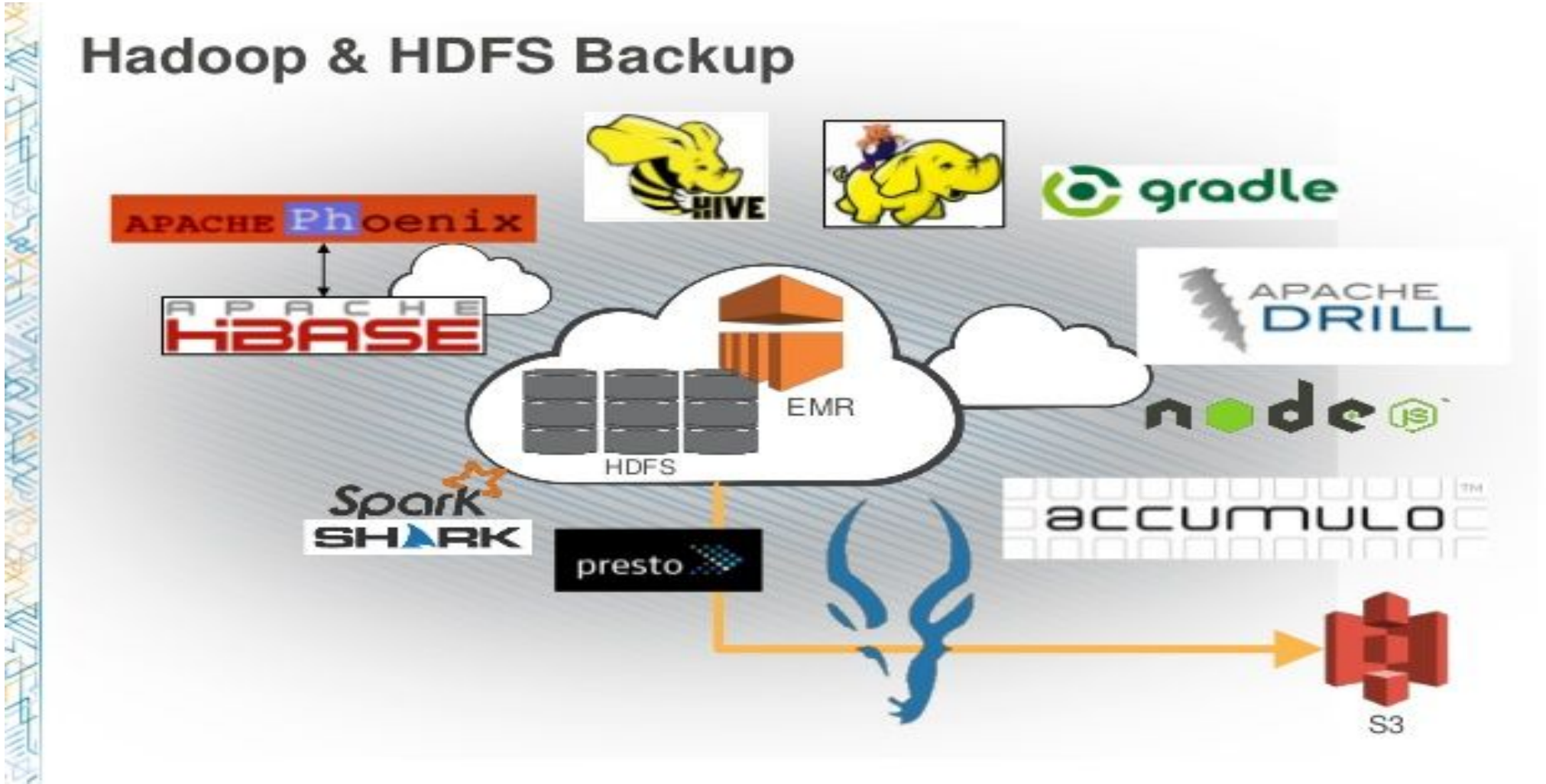
Aplicação

- Backup e recuperação
  - O S3 permite que o sistema de arquivos seja utilizado em políticas de backup
  - O sistema é escalável e utiliza o controle de versão da ferramenta
  - É possível configurar regras e ciclo de vida dos arquivos de backup, definindo políticas de retenção



Aplicação

- Backup e recuperação



Aplicação

- Análise de Big Data
  - A infraestrutura do S3 possui integração com serviços de análise de dados. Ex: Hadoop

- Armazenamento em nuvem híbrida
  - Em locais onde existem sistemas de arquivos locais, a solução híbrida é uma possibilidade de aumento de espaço
  - A classificação do nível de importância dos dados que definirá se permanecem na estrutura original ou se serão enviados à nuvem

Conceitos

- Buckets
  - É o objeto que realiza o agrupamento dos arquivos armazenados. Todos os arquivos estão inseridos em buckets

- Objetos
  - São as estruturas responsáveis pelo armazenamento dos metadados

Conceitos

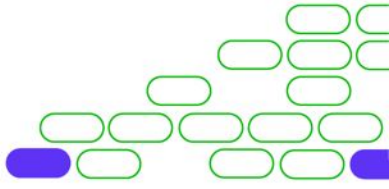
- Chaves
  - É o valor que identifica um objeto em um bucket.
  - Cada objeto no Amazon S3 pode ser endereçado através da combinação do endpoint de serviço da web, do nome de bucket, da chave. Por exemplo, a URL do *http://doc.s3.amazonaws.com/teste/AmazonS3.wsdl*
  - “doc” é o nome do bucket e “teste/AmazonS3.wsdl” é a chave.

- Regiões
  - Locais geográficos onde os buckets criados são armazenados. É possível o usuário definir esse locais manualmente

APIs do Amazon S3

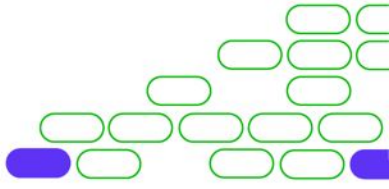
- Interface REST
  - Interface de programação para acesso das aplicações
  - Permite a utilização de requisições HTTP para criação, consulta e exclusão de objetos.

- Interface SOAP
  - Interface de programação que as aplicações podem realizar o acesso aos arquivos através de HTTPS, permitindo aplicações Java, .NET e outras se conectarem.

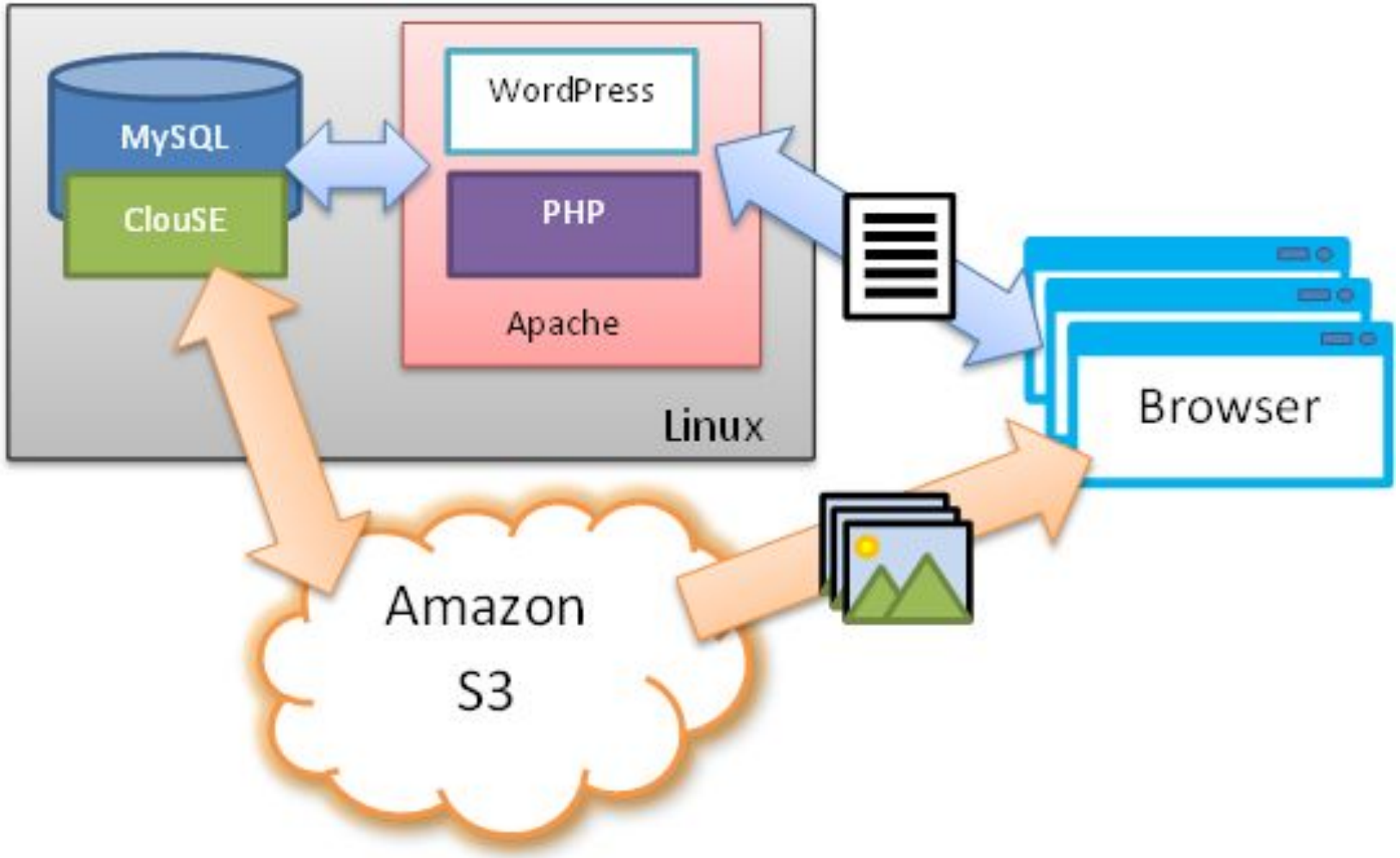


Estudo de Caso - Wordpress

- Em blogs e sites wordpress, à medida que o conteúdo se torna muito grande, uma solução possível é terceirizar a hospedagem de conteúdo
- O conceito de CDN (Content Delivery Network) pode ser implementado através da integração do site com o Amazon S3
- Através das urls do S3, os arquivos podem ser acessados pelos blogs
- O funcionamento do blog permanece o mesmo, de forma transparente

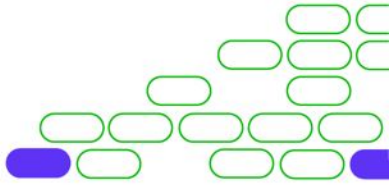


Estudo de Caso - Wordpress



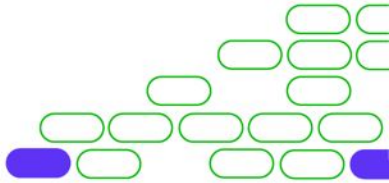
Estudo de Caso - Wordpress

- Justificativa
  - Baixo custo de contratação do S3
  - Redução do tamanho dos arquivos do site a serem transferidos
  - Contas com limitação de transferência de dados não são suspensas
  - Facilidade e simplicidade no uso da ferramenta
  - Integração com outros webservices da Amazon
  - Plugin nativo do wordpress na integração com o S3



Outros casos

- Netflix
  - O S3 é utilizado como “Data Lake” da Netflix
- Airbnb
  - Serviço de backup
- Thomsom Reuters
  - Utilização de análise de dados com Big Data e replicação



Conclusão

- ☑ O Amazon S3 é uma solução, em nuvem, disponibilizada como serviço de armazenamento de dados
- ☑ Possui recursos de integração com sites e com soluções da Amazon
- ☑ Fornece facilitadores na atividade de aumento de armazenamento
- ☑ Possui alta disponibilidade
- ☑ Pode ser utilizado como espaço de backup e armazém de dados para Big Data.

Faculdade XPe

Aula 5.4.1 – Google Cloud Storage

Nesta aula

- Google Cloud Storage e serviços correlatos

Introdução

- Google Cloud Storage, GCS, é uma solução de armazenamento presente na solução de nuvem da google.

- O google cloud storage surgiu para ser um componente de armazenamento concorrente ao Amazon S3, das soluções Amazon Webservices.

- O GCS é uma solução oferecida como serviço, com as vantagens de aplicativos hospedados em nuvem

Visão Geral

- A plataforma de serviços em nuvem da google, o Google Cloud Platform, opera de forma redundante, em diversos datacenters, em várias regiões do mundo.

- O objetivo do produto é fornecer toda a solução de infraestrutura e plataforma de uma empresa como serviço.

- A solução permite o desenvolvimento e integração de novos softwares às soluções da Google.

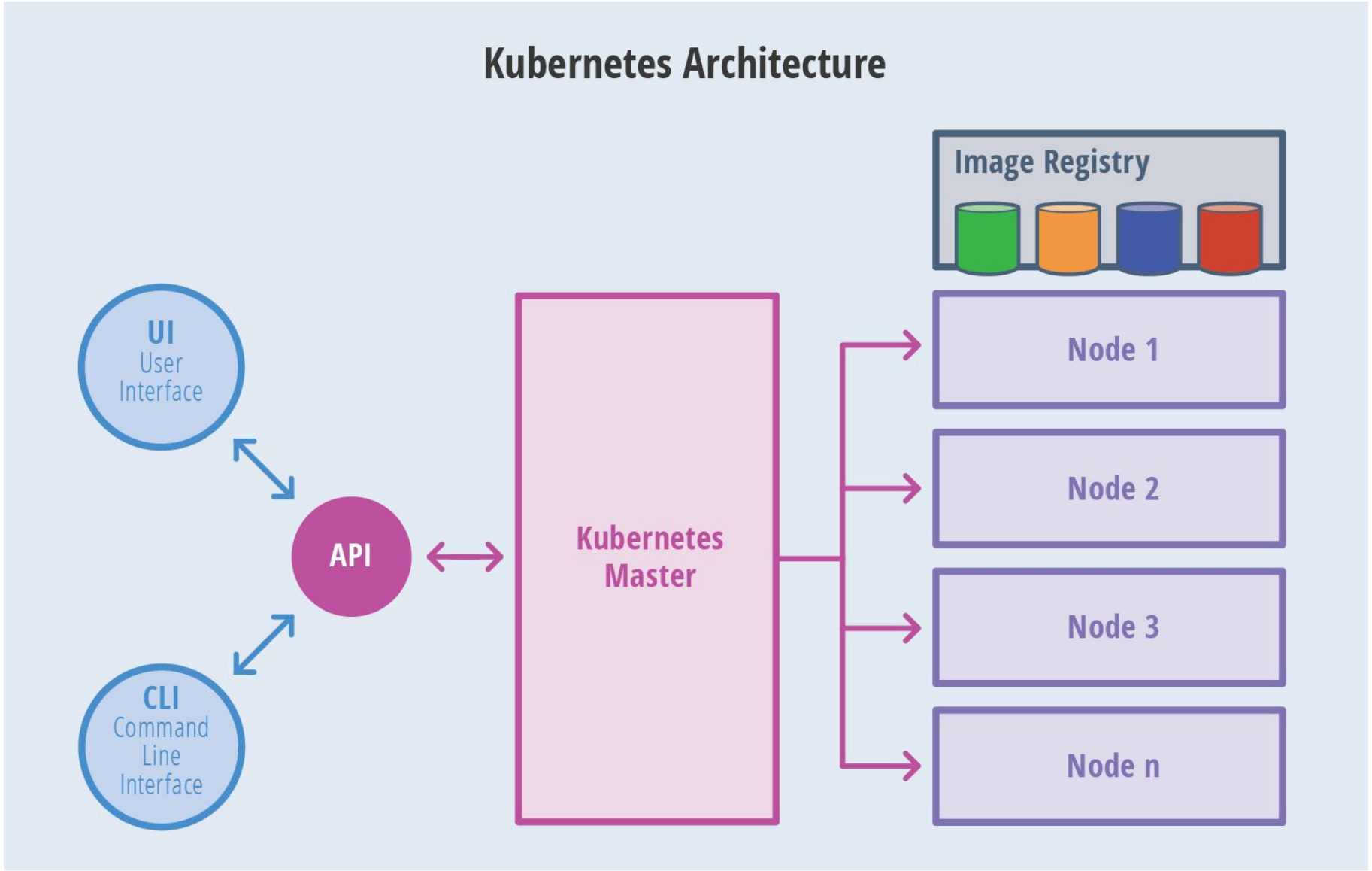
Google Cloud Platform - Serviços

COMPUTE	STORAGE AND DATABASES	NETWORKING	BIG DATA AND IoT	MACHINE LEARNING
Compute Engine	Cloud Storage	Virtual Private Cloud (VPC)	BigQuery	Cloud Machine Learning Engine
App Engine	Cloud SQL	Cloud Load Balancing	Cloud Dataflow	Cloud Jobs API
Container Engine	Cloud Bigtable	Cloud CDN	Cloud Dataproc	Cloud Natural Language API
Cloud Functions	Cloud Spanner	Cloud Interconnect	Cloud Datalab	Cloud Speech API
	Cloud Datastore	Cloud DNS	Cloud Dataprep	Cloud Translation API
	Persistent Disk		Cloud Pub/Sub	Cloud Vision API
	Data Transfer		Genomics	Cloud Video Intelligence
			Google Data Studio	
			Cloud IoT Core	

Compute

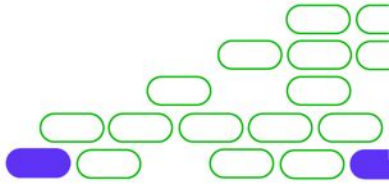
- Compute Engine
  - Serviço de virtualização que disponibiliza máquinas virtuais configuráveis e escaláveis para diferentes tipos de usuários.
  - Serviço e IaaS cobrado por tempo de utilização.
  - Integração com a solução Google Kubernetes para gerenciamento de aplicativos baseados em microsserviços

Compute


Kubernetes Architecture
Image Registry
UI
User
Interface
API
CLI
Command
Line
Interface
Kubernetes
Master
Node 1
Node 2
Node 3
Node n

Compute

- App Engine
  - Motor de criação de aplicações web.
  - Disponibiliza tecnologia de banco de dados NoSQL.
  - Possui API de autenticação dos usuários da Google.
  - A engine gerencia o trafego e escalona os recursos de acordo com o uso
- Ferramentas
  - Eclipse, IntelliJ, Maven, Git, Jenkins e PyCharm



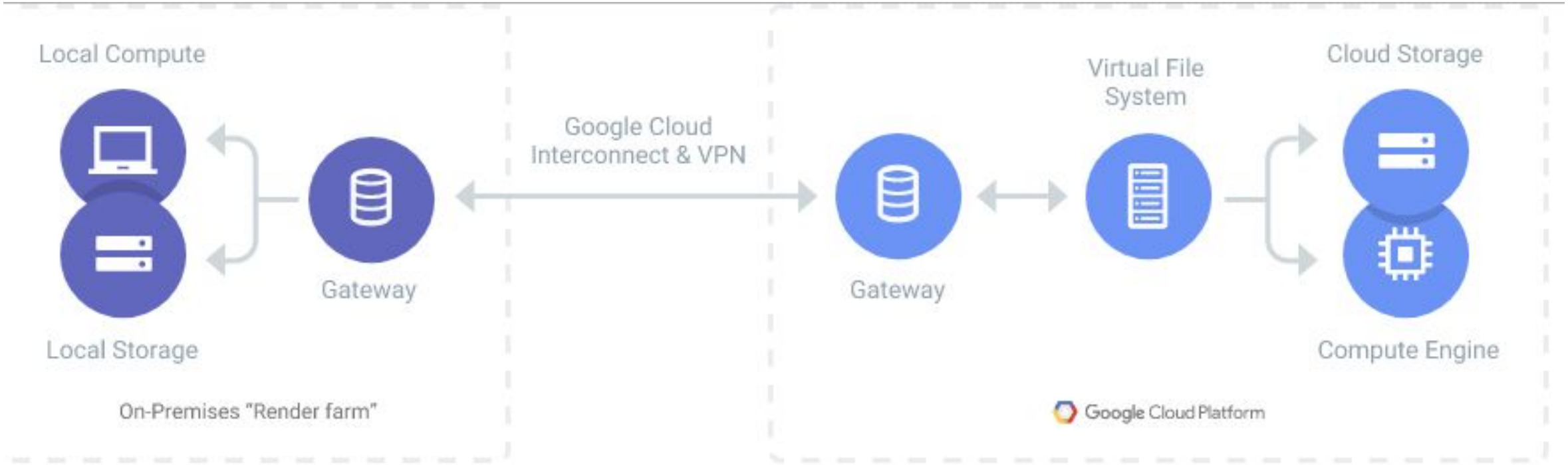
Networking

- Load Balance
  - Componente de balanceamento de carga, que realiza o balanceamento das aplicações criadas no Google Engine.
  - Escalonamento automático, de acordo com a demanda.

- Interconnect
  - Disponibiliza opções de interconexão entre a rede privada dos clientes com a infraestrutura em nuvem da google.

Networking

- Interconnect



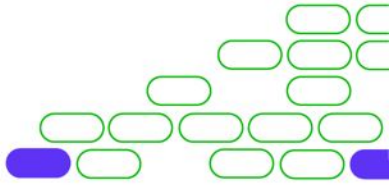
Big Data e IoT

- Big Query
  - Serviço de armazenamento de dados, Big Data, com suporte à linguagem SQL para análise dos dados

- Data Flow
  - Ferramenta de processamento de dados, tipo streaming
    - Possui algoritmos de detecção de comportamentos suspeitos
    - Possui análise de sensores, tecnologia IoT

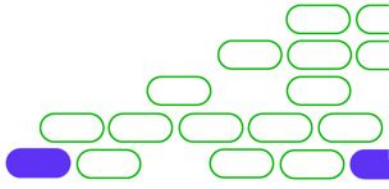
Big Data e IoT

- Cloud Pub/Sub
  - Serviço de comunicação entre as aplicações construídas pelo usuário no Google Cloud Engine.
  - Comunicação em tempo real.
  - Serviço de mensagens escalável, facilitando a integração entre aplicações e serviços até as ferramentas de análise.



Machine Learning

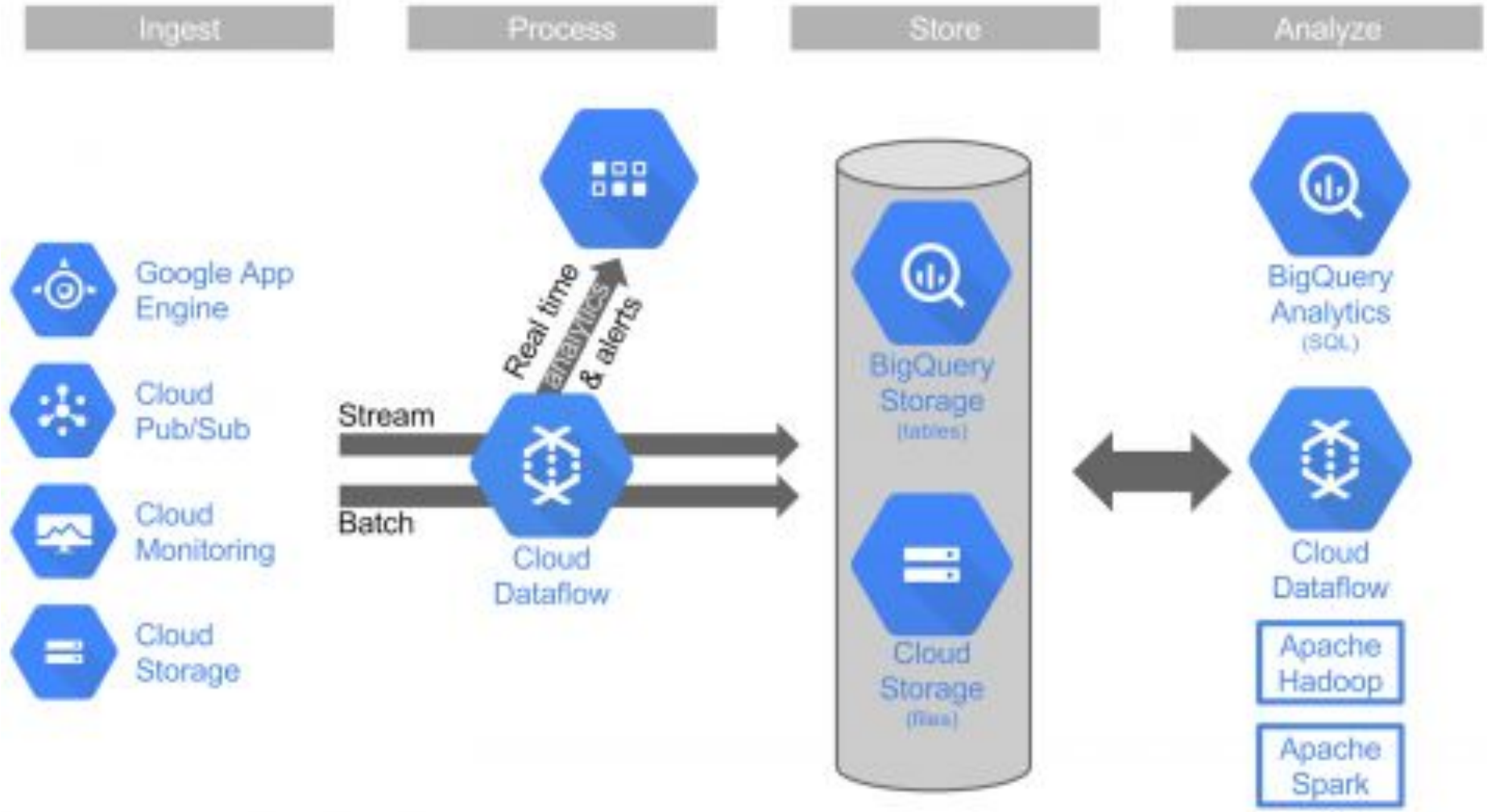
- Prediction
  - Serviço que permite a criação de soluções em aprendizado de máquina.
  - Possui a biblioteca TensorFlow para reúso dos modelos matemáticos e produtividade no desenvolvimento.
  - Através dos modelos de aprendizagem é possível construir soluções de análises preditivas.



Storage

- Solução de armazenamento da google com as seguintes características:
  - Regionalização / Multirregionalização
  - Definição de ciclos de vida de arquivos
    - Facilita a configuração de retenção das informações
  - Escalabilidade
    - Aumento de espaço sob demanda
  - Ferramentas de backup de máquina local para nuvem

Storage – Possível Solução



Conclusão

- ☑ As soluções de nuvem da Google entregam serviços para grande parte dos níveis corporativos
- ☑ Todas as soluções utilizam o google cloud storage para persistência de seus dados
- ☑ O google cloud storage pode ser utilizado como servidor de arquivos, mas também servidor de backup
- ☑ A escalabilidade e elasticidade permitem uso mais racional dos recursos