

Banco de dados

Engenharia de Telecomunicações

Prof. Emerson Ribeiro de Mello

mello@ifsc.edu.br

Licenciamento



Slides licenciados sob [Creative Commons "Atribuição 4.0 Internacional"](https://creativecommons.org/licenses/by/4.0/)

Sumário

- 1 Apresentação da disciplina
- 2 Introdução aos sistemas de banco de dados
- 3 Conceitos

Apresentação da disciplina

Relação com outras unidades curriculares



Ementa e objetivos da disciplina

■ Ementa

- Sistemas de bancos de dados. Projeto de banco de dados. Álgebra relacional. Normalização. Linguagem SQL. Conceitos sobre sistemas de gerenciamento de banco de dados (SGBD). Desenvolvimento de aplicações que fazem uso de banco de dados. Bancos de dados não relacionais.

■ Objetivos

- Conhecer conceitos fundamentais sobre organização da informação
- Projetar e modelar banco de dados relacional
- Usar ferramentas livres de SGBD para criar e manter banco de dados
- Desenvolver aplicações que façam uso de banco de dados

Metodologia

- Estudos serão guiados por leituras, exercícios e projetos
- Conteúdo será apresentado por meio de aulas expositivas e aulas práticas

Instrumentos de avaliação

Atividade	Quantidade	Peso	Recuperação
Lista de exercício (e)	2	10%	Dedução de 10% do valor total da nota para cada dia de atraso após o prazo de entrega, sendo 5 dias o prazo máximo
Avaliação escrita (a)	1	50%	Avaliação escrita substitutiva no final do semestre
Projeto prático (p)	1	40%	Após correção, o aluno terá 5 dias ou até o último dia do semestre letivo (o que ocorrer primeiro) para fazer nova entrega

- Critério para aprovação: mínimo 75% de presença e conceito final ≥ 6

$$CF = \left[a \times 0,5 + p \times 0,4 + \frac{\sum_{i=1}^2 e_i}{2} \times 0,1 \right], \quad CF \in \mathbb{N}. \quad (1)$$

Horários

■ **Aulas:** Laboratório de Redes de Computadores



- 07:30 – 09:20 - quarta-feira (quinzenal)
- 07:30 – 09:20 - sexta-feira

■ **Atendimento extraclasse:** Sala de Professores de Tele I



- 09:40 – 11:30 - quinta-feira



Possíveis interações do professor com a turma será por meio do SIGAA ou email (endereços obtidos no SIGAA)

Conteúdo programático

- 1 Conceitos sobre sistemas de bancos de dados
- 2 Diagrama Entidade-Relacionamento (ER)
- 3 Bancos de dados relacionais
- 4 Linguagem SQL
- 5 Formas normais
- 6 Desenvolvimento de aplicações *web*
- 7 NoSQL

Conteúdo programático

Nessa disciplina será visto

- Como modelar banco de dados relacionais
- Como fazer consultas em bancos de dados
- Como desenvolver aplicações que fazem uso de banco de dados
- Laboratórios com SQLite, MySQL e Java

Não será abordado como construir ou mesmo gerenciar de sistemas de banco de dados

Bibliografia básica

-  HEUSER, Carlos Alberto. **Projeto de banco de dados**. 6. ed.: Bookman, 2008. Disponível em: <<https://app.minhabiblioteca.com.br/books/9788595157552>>.
-  SILBERSCHATZ ABRAHAM; KORTH, Henry F.; Sudarshan. **Sistema de banco de dados**. 6. ed.: Campus, 2012. Disponível em: <<https://app.minhabiblioteca.com.br/books/9788595157552>>.



Os livros estão disponíveis no aplicativo Minha biblioteca. Acesse o SIGAA e no menu superior escolha **Serviços externos** → **Minha biblioteca**.

Introdução aos sistemas de banco de dados

Banco de dados para uma cafeteira *espresso*

Requisitos

- Cada usuário possui um cartão de identificação
- É possível tirar café simples ou duplo
- No final de cada mês, deve-se indicar o valor total a pagar para cada usuário



Banco de dados para uma cafeteira *espresso*

Requisitos

- Somente usuários cadastrados poderão usar a cafeteira
- É possível tirar café simples ou duplo
- No final de cada mês, indicar o valor total a pagar para cada usuário

Exercício

- 1 Identifique quais informações precisa armazenar
- 2 Organize as informações levantadas em arquivos texto

Arquivos texto estruturados

■ CSV (*Comma-separated values*)

```
aluno:x:1000:1000:Aluno do IFSC:/home/aluno:/bin/bash
```

■ JSON

```
{"login":"aluno", "uid": 1000, "gid": 1000, "name":"Aluno do IFSC", "home":"/home/aluno", "shell":"/bin/bash"}
```

■ YAML

```
user:  
- login: aluno  
- uid: 1000  
- gid: 1000  
- name: Aluno do IFSC  
- home: /home/aluno  
- shell: /bin/bash
```


Banco de dados para uma cafeteira *espresso*

Quais informações armazenar?

- Do usuário?
- Da cafeteira?
- Do consumo?
- Da fatura?

Como organizar as informações em arquivos texto?

- O que será armazenado em cada linha?
- Qual a estrutura de cada linha?
- Um ou vários arquivos?

Banco de dados para uma cafeteira *espresso*

Crie um aplicativo (i.e. em bash, Java, C) para interagir com os dados

Aplicação a ser desenvolvida

- 1 Como adicionar um novo usuário?
- 2 Como adicionar um novo consumo?
- 3 Como gerar a fatura?
- 4 Como excluir um usuário?

■ Operações típicas com arquivos texto

- 1 **adicionar** linha
- 2 **buscar** por informações
- 3 **alterar** informações em uma linha
- 4 **excluir** linha

Trecho em Java para trabalhar com CSV

```
/* ----- */
/*   Escrevendo no arquivo   */
/* ----- */
FileWriter arquivo = new FileWriter("usuarios.csv");
List<List<String>> linhas = new ArrayList<>();
linhas.add(Arrays.asList("123", "juca", "j@email"));

for(List<String> elem : linhas){
    arquivo.append(String.join(",", elem));
    arquivo.append("\n");
}
```

```
/* ----- */
/* Lendo conteúdo do arquivo */
/* ----- */
File entrada = new File("usuarios.csv");
Scanner linha = new Scanner(entrada);

while(linha.hasNext()){
    String[] registro = linha.nextLine().split(",");
    System.out.print(registro[0]);
}
```

Um ou vários arquivos?

■ Um único arquivo CSV

```
123,Juca,juca@email,simples,2018-07-20,08:00
123,Juca,juca@email,simples,2018-07-21,09:00
345,Paula,paula@email,,
567,Anna,anna@email,,
123,Juca,juca@email,duplo,2018-07-27,07:30
```

■ Vários arquivos CSV

```
123,Juca,juca@email
345,Paula,paula@email
567,Anna,anna@email
```

```
123,simples,2018-07-20,08:00
123,simples,2018-07-21,09:00
123,duplo,2018-07-27,07:30
```

- Como inserir ou fazer consultas em cada abordagem?
- Como garantir a integridade quando usar vários arquivos?

Banco de dados para uma cafeteira *espresso*

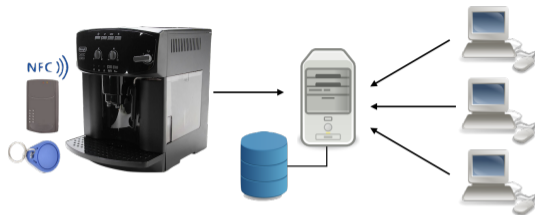
Novos requisitos

- 1 Cada usuário possui um saldo e só poderá tomar café se houver saldo na conta
- 2 No final de cada mês, deve-se indicar o total de cafés (e de qual tipo) que cada usuário consumiu
- 3 O histórico de consumo nunca poderá ser perdido
- 4 O usuário poderá ver a qualquer momento um extrato sobre seu consumo

Exercício

Crie uma nova versão do modelo de dados que criou anteriormente para atender esses novos requisitos

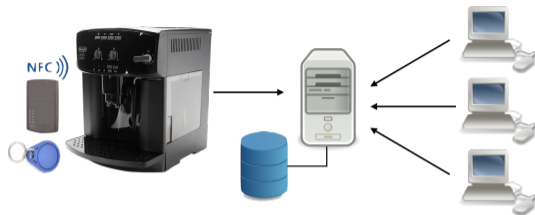
Compartilhar os arquivos csv em um servidor de arquivos



■ Acesso concorrente por múltiplos usuários

- Usuário colocando crédito no mesmo instante que um novo usuário é inserido na base

Compartilhar os arquivos csv em um servidor de arquivos



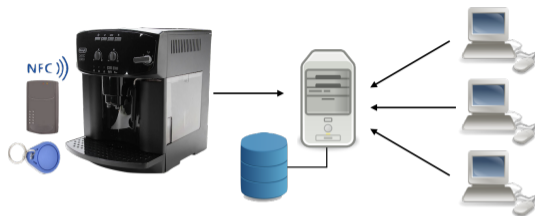
■ Acesso concorrente por múltiplos usuários

- Usuário colocando crédito no mesmo instante que um novo usuário é inserido na base

■ Atomicidade das atualizações

- Debitando crédito & adicionando registro de consumo

Compartilhar os arquivos csv em um servidor de arquivos



■ Acesso concorrente por múltiplos usuários

- Usuário colocando crédito no mesmo instante que um novo usuário é inserido na base

■ Atomicidade das atualizações

- Debitando crédito & adicionando registro de consumo

■ Controle de acesso

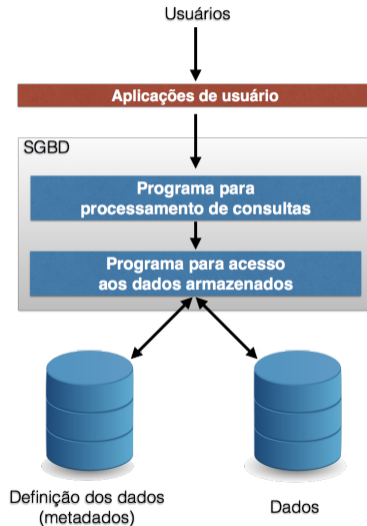
- Como garantir que somente parte dos dados esteja disponível para determinados usuários?

Conceitos

Conceitos sobre banco de dados

- **Banco de dados** é uma coleção de dados inter-relacionados
- **Sistema de gerenciamento de banco de dados** (SGBD) é um conjunto de programas que permitem aos usuários acessar e modificar esses dados
- **Dicionário de dados** contém metadados (dados sobre dados)
 - Estrutura dos dados, referencial de integridade, restrições, autorização, etc

Conceitos sobre banco de dados



Modelo de dados

- **Modelo de dados** é coleção de ferramentas para descrever os **dados**, seus **relacionamentos**, **semântica** e **restrições de consistência**
- **Entidade-relacionamento (ER)**
 - Baseado na percepção do mundo real que consiste em uma coleção de objetos (entidades) e os relacionamentos entre esses objetos
- Modelo relacional
- Baseado em objetos
- Semi-estruturado

Modelo de dados

- **Modelo de dados** é coleção de ferramentas para descrever os **dados**, seus **relacionamentos**, **semântica** e **restrições de consistência**
- Entidade-relacionamento (ER)
- **Modelo relacional**
- Baseado em objetos
- Semi-estruturado
- Coleção de tabelas para representar dados e os relacionamentos entre eles
- Cada tabela contém registros de um mesmo tipo
- Cada registro define um número fixo de campos ou atributos
- Modelo mais usado atualmente

Modelo de dados

- **Modelo de dados** é coleção de ferramentas para descrever os **dados**, seus **relacionamentos**, **semântica** e **restrições de consistência**
- Entidade-relacionamento (ER)
- Modelo relacional
- Baseado em objetos
- Semi-estruturado
- Pode ser visto como uma extensão do modelo ER com noções de encapsulamento e identidade do objeto

Modelo de dados

- **Modelo de dados** é coleção de ferramentas para descrever os **dados**, seus **relacionamentos**, **semântica** e **restrições de consistência**
- Entidade-relacionamento (ER)
- Modelo relacional
- Baseado em objetos
- **Semi-estruturado**
- Permite que registros do mesmo tipo possuam diferentes atributos – oposto daquilo que existe nos outros modelos
- Exemplos: XML e JSON

Modelo de dados

- **Modelo de dados** é coleção de ferramentas para descrever os **dados**, seus **relacionamentos**, **semântica** e **restrições de consistência**
- Entidade-relacionamento (ER)
- Modelo relacional
- Baseado em objetos
- Semi-estruturado

Esquema

Descrição de uma coleção de dados em particular com base em um modelo de dados

Exemplo de banco de dados relacional

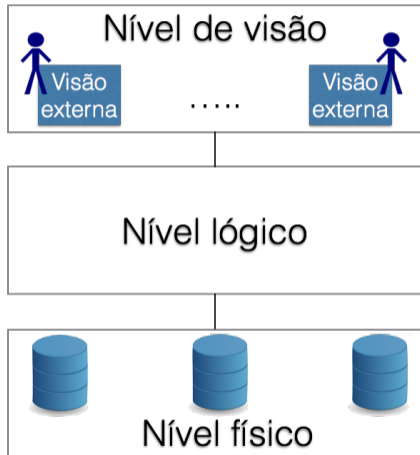
Tabela Alunos

Id	Nome	Curso
123	Juca	Telecomunicações
456	Amélia	Elétrica
789	Breno	Computação
900	Jucé	Automação

Tabela Cursos

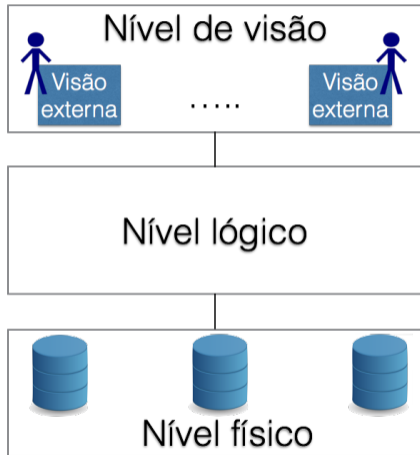
Curso	Campus	Orçamento
Telecomunicações	São José	100.000,00
Elétrica	Florianópolis	300.000,00
Computação	Lages	20.000,00
Automação	Chapecó	150.000,00

Níveis de abstração e independência de dados



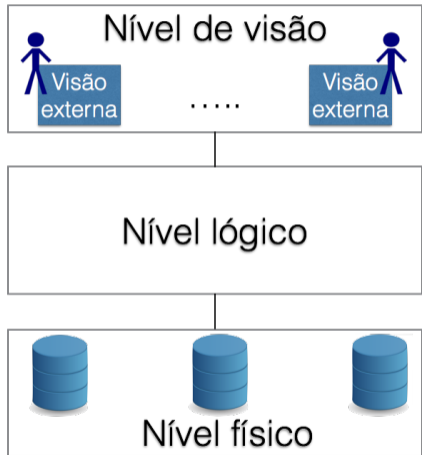
- **Nível físico** – Descreve como os dados são armazenados em baixo nível
 - Tabelas consistem em um conjunto de arquivos, cujo conteúdo não possui qualquer ordenação
 - Índices são dados específicos que possuem uma ordenação

Níveis de abstração e independência de dados



- **Nível lógico** – Descreve quais dados são armazenados e os relacionamentos entre eles
 - Ex: Tabela Aluno possui id, nome do aluno e nome do curso

Níveis de abstração e independência de dados



- **Nível de visão** – esquema externo que descreve a parte do banco de dados que um grupo de usuários pode ver
 - Ex: O coordenador de curso só consegue ver informações sobre os alunos de seu curso

Linguagem de definição de dados

Data Definition Language – DDL

- Linguagem para especificação do esquema ou estrutura do banco de dados
- Compilador DDL gera um conjunto de tabelas em um conjunto de arquivos denominado **dicionário de dados**

```
CREATE TABLE Alunos(  
  Id      INT AUTO_INCREMENT,  
  Nome   VARCHAR(60) not null,  
  Curso  VARCHAR(60) null,  
  PRIMARY KEY(Id)  
);
```

Linguagem de manipulação de dados

Data Manipulation Language – DML

- Permite acessar ou manipular dados no banco de dados
 - Recuperação, Inserção, Exclusão ou Modificação
- SQL (*Structured Query Language*) é a linguagem de consulta mais utilizada atualmente

```
SELECT Nome
FROM Alunos
WHERE Alunos.Id = '123';
```

```
SELECT Alunos.Nome, Alunos.Curso, Cursos.Campus
FROM Alunos, Cursos
WHERE Alunos.Curso = Cursos.Curso
AND Cursos.Curso = "Telecomunicacoes";
```

Algum problema com esse projeto de banco de dados?

Id	Nome	Curso	Disciplina	Professor
123	Juca	Telecomunicações	Sinais	João
456	Amélia	Elétrica	Sinais	João
789	Breno	Computação	Programação I	Martin
900	Jucé	Automação	Programação I	Martin
334	Maira	Telecomunicações	Sistemas Distribuídos	Paulo
453	Célio	Telecomunicações	Cálculo II	Luíza
112	Cícero	Computação	Cálculo II	Luíza
322	Marco	Automação	Cálculo II	Luíza
567	Alonso	Computação	Sistemas Distribuídos	Paulo
257	Luiz	Telecomunicações	Sinais	João

Algum problema com esse projeto de banco de dados?

Id	Nome	Curso	Disciplina	Professor
123	Juca	Telecomunicações	Sinais	João
456	Amélia	Elétrica	Sinais	João
789	Breno	Computação	Programação I	Martin
900	Jucé	Automação	Programação I	Martin
334	Maira	Telecomunicações	Sistemas Distribuídos	Paulo
453	Célio	Telecomunicações	Cálculo II	Luíza
112	Cícero	Computação	Cálculo II	Luíza
322	Marco	Automação	Cálculo II	Luíza
567	Alonso	Computação	Sistemas Distribuídos	Paulo
257	Luiz	Telecomunicações	Sinais	João

Normalização

Criar um conjunto de tabelas que permita armazenar informações sem redundância desnecessária

Diagramas E-R (notação baseada na UML)



Figura: Diagrama E-R

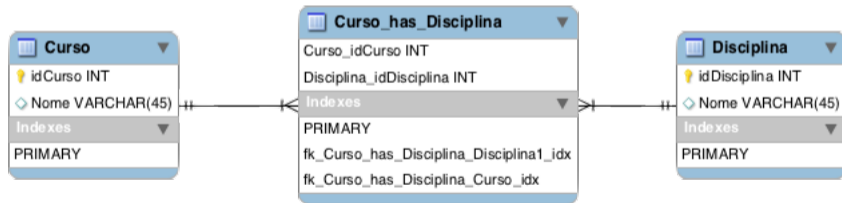


Figura: Diagrama E-R gerado pela ferramentas MySQL Workbench

Banco de dados transacional – propriedades ACID

Garante que todas operações de consulta ou de alteração são **atômicas**, **consistentes**, **isoladas** e **duráveis**.

■ **A**tomicidade

- Todas operações (leitura/escrita) em uma transação são executadas com sucesso ou tudo é desfeito

■ **C**onsistência

- A execução de uma transação leva o banco de um estado consistente para um outro estado consistente

■ **I**solamento

- Transações podem acontecer de forma concorrente sem qualquer interferência

■ **D**urabilidade

- Ao concluir uma transação, todas modificações geradas serão persistentes

Transação

```
read saldoContaOrigem
write (saldoContaOrigem - 500)
read saldoContaDestino
write (saldoContaDestino + 500)
```

■ Antes

- Saldo origem: 600,00
- Saldo destino: 300,00

■ Depois

- Saldo origem: 100,00
- Saldo destino: 800,00

Transação

```
read saldoContaOrigem
write (saldoContaOrigem - 500)
read saldoContaDestino
write (saldoContaDestino + 500)
FALHA NO SISTEMA
```

■ Antes

- Saldo origem: 600,00
- Saldo destino: 300,00

■ Depois

- Saldo origem: 100,00
- Saldo destino: 800,00

Transação

```
read saldoContaOrigem  
write (saldoContaOrigem - 500)
```

FALHA NO SISTEMA

```
read saldoContaDestino  
write (saldoContaDestino + 500)
```

■ Antes

- Saldo origem: 600,00
- Saldo destino: 300,00

■ Depois

- Saldo origem: 100,00
- Saldo destino: 300,00

Transação

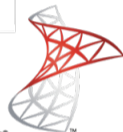
- SGBD garantem que a execução concorrente das transações T_1, \dots, T_n seja equivalente a uma execução serial dessas transações
 - Antes de ler ou escrever, a transação obtém acesso a uma seção crítica e só libera depois que for concluída
- A atomicidade e durabilidade podem ser garantidas por meio de um arquivo de registro (*log*)
 - Mantenha no *log* todas as escritas que foram concluídas
 - Se houver alguma falha, então desfaça as escritas parciais (que não foram escritas no *log*)

Alguns SGBDs relacionais



PostgreSQL

ORACLE®
DATABASE



Microsoft®
SQL Server™



SQLite



MySQL®

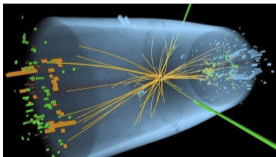


MariaDB



Big data

- Grande **volume** de dados
 - gerados em grande **velocidade**
 - e com grande **variedade**
 - que exigem novas formas de processamento para ajudar nas tomadas de decisão (alguns ainda citam **veracidade** e **valor**)
-
- Fundamentado sobre tecnologias de análise (Hadoop / MapReduce) e infraestrutura de armazenamento e processamento



Instrumentos científicos



Redes sociais



Cidades inteligentes



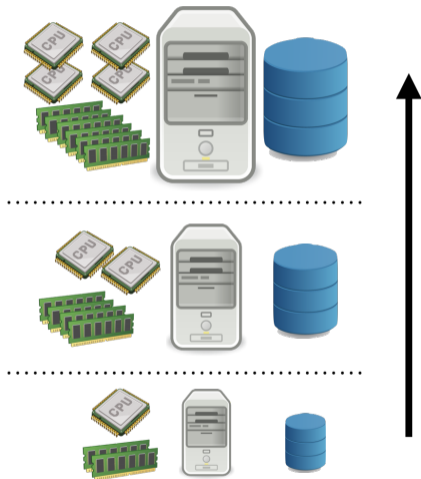
Dados de telefones

Banco de dados NoSQL

Adequado para armazenar uma vasta quantidade de dados de maneira efetiva, com baixo custo e com facilidade para implementar escalonamento horizontal (distribuição por diversos nós)

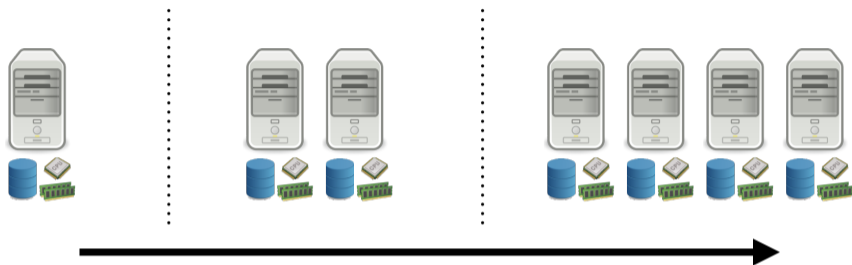
- Modelo de dados não é baseado em tabelas
 - Em uma tabela, todas as linhas terão o mesmo número de colunas
- Estrutura de dados em bancos NoSQL
 - chave-valor
 - colunas esparsas
 - grafo
 - orientado a documento

Escalonamento vertical



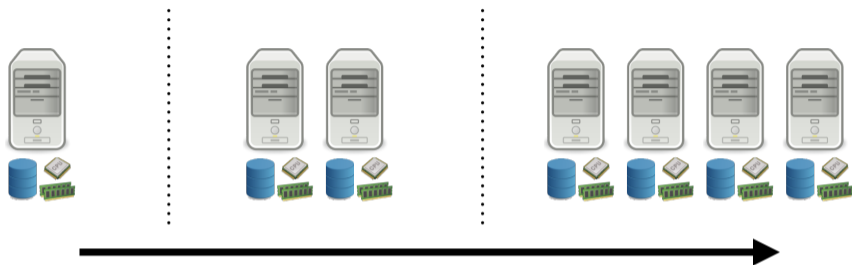
Adiciona mais recursos (memória, cpu, disco) em um único nó para permitir que atenda o aumento da demanda

Escalonamento horizontal



- Capacidade de processamento e armazenamento é aumentada por meio de adição de novos nós no *cluster*

Escalonamento horizontal

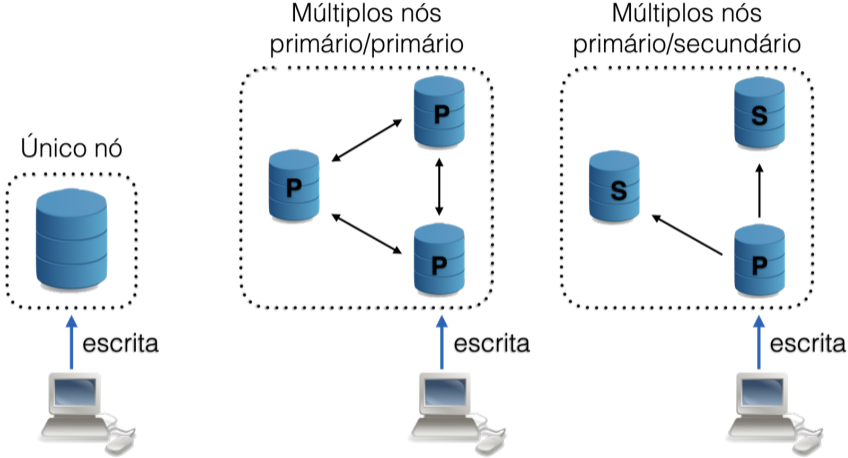


- Capacidade de processamento e armazenamento é aumentada por meio de adição de novos nós no *cluster*

Escalabilidade horizontal é a opção mais barata e flexível, porém possui uma maior complexidade

- Sistema centralizado vs sistema distribuído

Sincronismo das réplicas



Teorema de CAP

Consistency, Availability and Partition tolerance

- **Consistência** – Toda operação de leitura em qualquer nó do *cluster* deve retornar a última escrita ou um erro
- **Disponibilidade** – Todo nó não falho retornará uma resposta dentro um limite de tempo razoável, porém não há garantia que se refere a escrita mais recente
- **Tolerância a partição** – o sistema continua a funcionar mesmo diante de mensagens perdidas/atrasadas ou falhas em alguns nós

Em um sistema de armazenamento de dados distribuídos é impossível oferecer simultaneamente mais de duas das garantias acima

- Diante do particionamento da rede, é necessário escolher entre consistência e disponibilidade

Banco de dados NoSQL estão fundamentados sobre modelo de consistência eventual (BASE)

■ **Basic Availability**

- Todo pedido terá uma resposta, porém a resposta pode indicar uma falha na tentativa de obter o dado ou que o dado retornado está em um estado inconsistente

■ **Soft-state**

- O estado do sistema pode alterar ao longo do tempo, mesmo durante intervalo de tempo que não houve qualquer escrita

■ **Eventual consistency**

- Sistema se tornará consistente ao longo do tempo uma vez que não se tenha novas operações de escrita

ACID vs BASE

- As propriedades do BASE são menos restritivas que as garantias do modelo ACID
- Para o BASE pode ser mais importante garantir a disponibilidade dos dados (requisito para ambientes de larga escala) do que oferecer a garantia da consistência dos dados em todas as réplicas

ACID vs BASE

- As propriedades do BASE são menos restritivas que as garantias do modelo ACID
- Para o BASE pode ser mais importante garantir a disponibilidade dos dados (requisito para ambientes de larga escala) do que oferecer a garantia da consistência dos dados em todas as réplicas

Qual modelo seria mais adequado?

- Para armazenar dados das contas bancárias de uma instituição financeira
- Para armazenar os *posts* na linha do tempo em uma rede social

Alguns bancos de dados NoSQL

APACHE
HBASE



Cassandra



CouchDB
relax



riak



mongoDB

HYPERTABLE^{INC}



Neo4j



redis