

## ***Estilos de Arquiteturas de Dados***

### ***Arquitetura de três esquemas ANSI/SPARC***

A arquitetura de três esquemas é uma maneira de organizar bancos de dados em três partes para facilitar o uso e o gerenciamento:

1. **Camada Interna:** Define como os dados são fisicamente armazenados no sistema.
2. **Camada Conceitual:** Descreve a estrutura lógica e as relações entre os dados.

**Exemplo:** Cliente e Conta são tipos de dados; cada cliente pode ter várias contas.

3. **Camada Externa:** Mostra os dados de uma forma personalizada para cada usuário.

**Exemplo:** Um cliente vê só suas contas; o gerente vê as contas de todos.

### ***Categoria Dos comandos SQL***

- **DDL:** CREATE, ALTER, DROP — estrutura do banco.
- **DML:** INSERT, UPDATE, DELETE — manipulação de dados.
- **DCL:** GRANT, REVOKE — controle de acesso.
- **TCL:** COMMIT, ROLLBACK, SAVEPOINT — gerenciamento de transações.
- **DRL:** SELECT — consulta dados.

## ***Arquiteturas Centralizadas***

### ***1. Arquitetura Centralizada (ou de Uma Camada)***

- **Como funciona:** Todos os dados e processos estão em um único servidor. Os usuários se conectam a esse servidor para acessar tudo.
- **Exemplo:** Um banco de dados que está em um único computador. Quando você acessa, tudo é feito lá.

### ***2. Arquitetura de Duas Camadas (Dual Tier)***

- **Como funciona:** Dividido em duas partes: a camada do cliente (onde os usuários interagem) e a camada do servidor (onde os dados estão armazenados). O cliente se conecta diretamente ao servidor.
- **Exemplo:** Um aplicativo de desktop que envia solicitações diretamente para um banco de dados em um servidor.

### ***3. Arquitetura em Três Camadas (Three-Tier)***

- **Como funciona:** Tem três partes:
  1. **Camada de Apresentação** (interface do usuário),
  2. **Camada de Lógica de Negócio** (processa as regras e a lógica),
  3. **Camada de Dados** (armazenamento dos dados). As camadas se comunicam, mas cada uma tem uma função específica.
- **Exemplo:** Um site onde a interface é separada do servidor que processa pedidos e do banco de dados que armazena informações.

## ***Normalização de Dados***

### Modelo Conceitual

O modelo Conceitual no DER (Diagrama de Entidade-Relacionamento) é a primeira etapa de modelagem de um banco de dados. Ela foca na identificação das entidade, relacionamentos e atributos.

Usando:

- Retângulos Para entidades
- Losangos para relacionamentos
- Círculos para atributos

**Modelagem Conceitual:** Fundamental para identificar e definir relações, de forma que sejam claramente definidas, e economiza tempo, porque ajuda a compreender a estrutura do banco de dados.

**Modelo Lógico:** O modelo lógico é uma etapa intermediária na criação de banco de dados, onde as entidades identificadas no modelo conceitual são transformadas em tabelas e colunas.

Essa fase traduz o entendimento conceitual, próximo a visão de um profissional de banco de dados, ajuda porque prepara o terreno para a implementação física do banco de dados

**Modelo Físico** - > É a implementação direta no banco de dados, detalhando como as informações serão armazenadas e organizadas.

## ***Padrão de Projetos***

Um padrão de sistema é como um "passo a passo" para resolver problemas comuns ao construir sistemas, tornando-os melhores e mais seguros.

Esses padrões são desenvolvidos pela comunidade de engenheiros e desenvolvedores com base em suas experiências e melhores práticas.

Eles fornecem soluções para problemas comuns enfrentados durante o desenvolvimento e manutenção de sistemas.

As vantagens dos padrões de projeto são: solução rápida para problemas comuns, código mais organizado, fácil de manter e reutilizar.

## ***Sistemas Distribuídos***

Sistemas distribuídos são conjuntos de computadores que trabalham juntos e trocam informações via rede. Aqui vão os pontos principais:

1. **Sem Relógio Global:** Não há um único "relógio" para todos os computadores, então eles precisam se coordenar por mensagens.
2. **Execução Simultânea:** Vários processos rodam ao mesmo tempo, usando recursos compartilhados de forma eficiente.
3. **Concorrência:** Múltiplos sistemas acessam e usam recursos ao mesmo tempo, exigindo sincronização para evitar conflitos (como em um sistema de arquivos compartilhado).
4. **Falhas Independentes:** Se um computador falha, o sistema continua funcionando com os outros.

## **Padrões Comuns em Sistemas Distribuídos**

**Pipes e Filtros:** Processamento em etapas, onde cada etapa trata uma parte dos dados.

**Exemplo:** No Unix, comandos como grep, sort e awk são combinados para processar dados em sequência.

**Microkernel:** Mantém apenas o núcleo essencial do sistema, deixando o restante como módulos opcionais.

**Exemplo:** MINIX, onde só o básico roda no núcleo para ser mais flexível.

**ORB (Object Request Broker):** Permite que partes de um sistema falem entre si, mesmo se forem diferentes.

**Exemplo:** CORBA, que conecta componentes feitos em diferentes linguagens de programação.

## Aula 2 - Modelagem de dados

MER: Representam como as entidades, atributos e relacionamentos estão distribuídos no sistema de informação.

Aqui estão explicações curtas para cada tipo de repositório de dados:

1. **Data Warehouse:** Um grande banco de dados que armazena informações para gerar relatórios e análises de negócios.
2. **Data Mart:** Um "mini data warehouse" focado em um departamento específico, como vendas ou marketing.
3. **Metadados:** Dados sobre os dados, como origem e descrição, ajudando a entender e gerenciar as informações.
4. **Cubo de Dados:** Estrutura que organiza dados em várias dimensões, útil para análises detalhadas.
5. **Data Lake:** Armazena dados brutos em diferentes formatos, prontos para serem processados depois.

### Exemplo dos serviços acima:

**Data Warehouse:** Exemplo - Amazon Redshift. Armazena dados grandes para relatórios de negócios.

**Data Mart:** Exemplo - Salesforce Analytics Cloud. Mini armazém de dados focado em um setor, como finanças.

**Data Lake:** Exemplo - Amazon S3. Guarda dados brutos de diferentes tipos, prontos para serem processados.

**Repositório de Código:** Exemplo - GitHub. Onde o código é salvo e gerenciado.

**Repositório de Clientes/CRM:** Exemplo - Salesforce CRM. Organiza informações de clientes para acompanhar o relacionamento com eles.

## Conceitos de arquitetura em nuvem

Aqui estão os modelos de serviços em nuvem de forma bem simples:

1. **IaaS (Infraestrutura como Serviço):** Oferece servidores e armazenamento pela internet, para que empresas possam construir e gerenciar suas próprias plataformas.
  - **Exemplo:** Amazon EC2.
2. **PaaS (Plataforma como Serviço):** Fornece uma plataforma pronta para desenvolver e hospedar aplicativos, sem se preocupar com a infraestrutura.

- **Exemplo:** Google App Engine.
- 3. **SaaS (Software como Serviço):** Aplica-se a softwares disponíveis na nuvem, que você usa sem precisar instalar ou manter.
  - **Exemplo:** Gmail, Office 365.

#### **Vantagens da Computação em Nuvem:**

- **Redução de Custos:** Menos gastos com equipamentos.
- **Escalabilidade:** Aumenta ou reduz recursos conforme a necessidade.
- **Acessibilidade:** Acessa dados de qualquer lugar com internet.

#### **Data Stewards:**

- **Definição:** Profissionais que gerenciam dados dentro de uma empresa.
- **Importância:** Garantem que os dados sejam seguros, precisos e usados corretamente, ajudando na tomada de decisões.

#### **Responsabilidades dos Data Stewards:**

1. **Monitorar Dados:** Acompanham dados desde a entrada até o armazenamento.
2. **Proteger Dados:** Mantêm as informações seguras contra ameaças.
3. **Apoiar TI:** Ajudam na atualização de sistemas e implementação de novos programas.

#### **Conhecimentos Necessários:**

- **Governança de Dados:** Saber como gerenciar dados.
- **Modelagem de Dados:** Organizar e estruturar dados.
- **Segurança de Dados:** Proteger e garantir a integridade das informações.