**Anotações - Banco de Dados**

**1. Ciclo de Desenvolvimento de um Banco de Dados**

1. **Modelar**: Definir a estrutura lógica dos dados.
   * **Exemplo prático**: Em uma empresa de logística, modela-se os dados como **Clientes, Pedidos, Produtos e Entregas**.
2. **Projetar**: Criar o esquema do banco, incluindo tabelas, chaves e normalização.
   * **Exemplo prático**: Criar tabelas como **Tabela\_Clientes (CPF, Nome, Endereço)** e **Tabela\_Pedidos (ID\_Pedido, CPF\_Cliente, Data, Status)**.
3. **Implementar**: Escrever os comandos SQL para criação de tabelas e inserção de dados.
   * **Exemplo prático**: Utilizar comandos **CREATE TABLE** e **INSERT INTO** para criar e popular as tabelas no banco de dados.
4. **Testar**: Validar as consultas, índices e integridade dos dados.
   * **Exemplo prático**: Executar consultas para garantir que um cliente possa ter vários pedidos sem duplicação de dados.
5. **Entregar**: Implantar o banco de dados no ambiente de produção.
   * **Exemplo prático**: Disponibilizar o banco para uso após a validação de todas as funcionalidades.
6. **Capacitar**: Treinar os usuários e administradores para operar e manter o sistema.
   * **Exemplo prático**: Realizar treinamentos para que os funcionários saibam como consultar e registrar pedidos.

**2. Modelo Conceitual**

**2.1 Modelo de Dados**

* **Definição**: Define como os dados são organizados, armazenados e manipulados no banco de dados.
* **Exemplo prático**: Um supermercado adota um modelo de dados relacional para armazenar informações de **produtos, fornecedores e vendas**, permitindo consultas eficientes.

**2.2 Entidades e Modelo ER**

* **Definição**: As entidades representam objetos do sistema, como **Cliente, Produto e Ordem de Serviço**, e as relações entre elas são modeladas com chaves primárias e estrangeiras.
* **Exemplo prático**: Em uma biblioteca, as entidades **Livros, Usuários e Empréstimos** são representadas, com chaves que ligam as tabelas entre si.

**2.3 Classes**

* **Definição**: Na programação orientada a objetos, as **classes** encapsulam dados e comportamentos.
* **Exemplo prático**: Em um sistema de vendas, a classe **Produto** tem atributos como **nome, preço e estoque**, e métodos para aplicar descontos.

**2.4 Categoria e Modelo**

* **Definição**:
  + **Categoria**: Agrupamento de entidades com características semelhantes.
  + **Modelo**: A estrutura de um objeto dentro de uma categoria.
* **Exemplo prático**: Em um marketplace, os produtos são classificados em categorias como **Eletrodomésticos, Roupas e Brinquedos**, e dentro de **Eletrodomésticos**, cada modelo de produto tem seus próprios atributos.

**2.5 Chaves Primárias e Secundárias**

* **Atributo Chave Primária**: Identificador único de uma entidade.
  + **Exemplo prático**: O **ID do cliente** é a chave primária na tabela **Clientes**, garantindo que cada cliente tenha um registro único.
* **Atributo Chave Secundária**: Usado para busca, mas não para garantir unicidade.
  + **Exemplo prático**: O **nome do cliente** pode ser uma chave secundária, usada para buscas rápidas, mas não garante unicidade.

**3. Sistemas e Dados**

**3.1 Adoção da Informática em Organizações**

* **Definição**: Implementação gradual de sistemas para áreas como **vendas, produção e compras**, onde os dados dos produtos são armazenados em um banco de dados.
* **Exemplo prático**: Uma empresa de varejo integra um sistema para controlar vendas, onde os dados de produtos e estoque são centralizados em um SGDB.

**3.2 Sistemas Isolados**

* **Definição**: Sistemas que não compartilham dados entre si, prejudicando a integração e a consistência.
* **Exemplo prático**: Um sistema de vendas que não se comunica com o sistema de estoque pode causar a venda de produtos sem estoque.

**3.3 Redundância Não Controlada**

* **Definição**: Duplicação de dados sem um controle adequado, levando a inconsistências e desperdício de armazenamento.
* **Exemplo prático**: O cadastro duplicado de um cliente com nomes ligeiramente diferentes (como **"João Silva" e "João da Silva"**) gera inconsistência nos dados.

**3.4 Validação de CPF**

* **Definição**: A validação de CPFs é feita por meio de uma **API integrada com gov.br**, não sendo responsabilidade do SGDB.
* **Exemplo prático**: Um sistema de fidelidade valida o CPF de clientes utilizando a API do gov.br para garantir que o cadastro seja correto.

**3.5 Ordem de Serviço (O.S)**

* **Definição**: Um registro único que documenta um serviço prestado, vinculado a um cliente e produto.
* **Exemplo prático**: Em uma assistência técnica, cada **Ordem de Serviço** contém detalhes do cliente, produto e defeito, com um **identificador único**.

**3.6 Relação Cliente x O.S no Modelo Relacional**

* **Definição**: No **modelo relacional**, as entidades **Cliente** e **Ordem de Serviço (O.S)** se relacionam por meio de **chaves primárias e estrangeiras**.
* **Exemplo prático**: Em uma clínica veterinária, um **cliente** pode ter vários **animais**, e cada **animal** pode ter várias **consultas** (O.S).

**3.7 Variáveis Necessárias para Celular**

* **Definição**:
  1. **CPF (Cliente)**: Identificação do cliente.
  2. **Campo de Integração**: Permite que sistemas se comuniquem entre si.
  3. **Produto (Marca e Modelo) e Defeito**: Informa o item e o defeito que precisa ser consertado.
* **Exemplo prático**: Em uma loja de celulares, o **CPF do cliente**, **dados do celular (marca, modelo)** e a **descrição do defeito** são essenciais para registrar o serviço.

**3.8 Origem dos Dados**

* **Definição**: Os dados podem vir de outras fontes além da **Ordem de Serviço**, como cadastros prévios de clientes.
* **Exemplo prático**: Em um hospital, a origem dos dados pode ser tanto do **cadastro do paciente** quanto dos **registros de consultas**.

**4. Chaves e Identificadores**

**4.1 Chaves Primárias**

* **Definição**: Identificador único de cada entidade no banco de dados, garantindo a integridade dos dados.
* **Exemplo prático**: O **CPF do cliente** ou o **ID do produto** garantem que cada registro seja único no sistema.

**4.2 Unicidade da Ordem de Serviço**

* **Definição**: Cada Ordem de Serviço deve possuir um **identificador único**.
* **Exemplo prático**: Em uma assistência técnica, cada **O.S** recebe um número único, garantindo que duas ordens não possam ser idênticas.