# BANCO DE DADOS ORIENTADOS A OBJETOS

**Fernando Fonseca**

## ORIENTAÇÃO A OBJETOS

- Origem

  Linguagens de Programação que incluíram Tipos Abstratos de Dados entre seus Conceitos
- Principal Vantagem

  O Objeto permite Modelar mais Aproximadamente os Objetos do Mundo Real



## ORIENTAÇÃO A OBJETOS

- Sistemas de Gerenciamento de Banco de Dados Orientado a Objetos (SGBDOO)

  Armazenam e Gerenciam o Acesso a um Conjunto de Objetos, permitindo o Compartilhamento deles por Várias Aplicações



## ORIENTAÇÃO A OBJETOS

- Novos Aplicativos
  - Projeto Assistido por Computador (CAD)

    Armazenar Dados pertencentes a um Projeto de Engenharia, incluindo os Componentes de um Item sendo projetado, o Inter-relacionamento de Componentes e Antigas Versões de projetos.



## ORIENTAÇÃO A OBJETOS

- Novos Aplicativos (Cont.)
  - Engenharia de Software Assistida por Computador (CASE)

    Armazenar Dados Requeridos para Auxiliar Desenvolvedores de Software, incluindo Código Fonte, Dependências entre Módulos de Software, Definições e Usos de Variáveis e a História do Desenvolvimento do Sistema.



## ORIENTAÇÃO A OBJETOS

- Banco de Dados Multimídia

  Armazenar Dados Espaciais, Dados de Áudio, Dados de Vídeo. Visam atender Aplicações Geofísicas, Artísticas, etc,
- Sistemas de Informação de Escritórios (OIS)

  Um BD para Automação de Escritórios, o qual deve incluir Ferramentas baseadas principalmente em Estações de Trabalho para Criação e Manutenção de Documentos, Agendas, etc.

## ORIENTAÇÃO A OBJETOS

- Novos Aplicativos (Cont.)
  - Sistemas de Banco de Dados Dedutivos

    Incluem não apenas Dados, mas também Regras Explícitas representando Restrições de Integridade, Gatilhos e Outros Conhecimentos sobre o Empreendimento Modelado.



## ORIENTAÇÃO A OBJETOS

- Requerimentos dos Novos Aplicativos
  - Objetos Complexos

    Um Item visto como um único Objeto no Mundo Real, mas que contém Outros Objetos, podendo ter uma Estrutura Interna Arbitrariamente Complexa.



## ORIENTAÇÃO A OBJETOS

- Requerimentos dos Novos Aplicativos (Cont.)
  - Dados Comportamentais

    Objetos Distintos podem precisar Responder de Modos Diferentes a um Mesmo Comando. Essa Informação Comportamental pode ser Capturada pelo Armazenamento de Código Executável com os Objetos do BD.



## ORIENTAÇÃO A OBJETOS

- Requerimentos dos Novos Aplicativos (Cont.)
  - Metaconhecimento

    Freqüentemente os Dados mais Importantes sobre os Aplicativos são as Regras Gerais em vez de Dados Particulares. Esse Conhecimento pode ser Armazenado em Forma de Regras, como Predicados de Primeira Ordem.



## ORIENTAÇÃO A OBJETOS

- Transações Longas

  Os Aplicativos de CAD e CASE envolvem a Interação Humana com os Dados. Algumas delas podem ser Modificações que Durem por um Período de Tempo Muito Maior do que as Realizadas em Sistemas Comerciais.
  - Os Esforços de Vários Projetistas Podem Gerar Conflitos que Levem a Abortos de Transações, Esperas por Bloqueios, etc, Muito mais Sérios que em Transações Curtas.



## ORIENTAÇÃO A OBJETOS

- Transações Longas (Cont.)
  - Um Novo Modelo de Transação é Necessário para dar Suporte a essas Aplicações.

## ORIENTAÇÃO A OBJETOS

- Estrutura do Objeto

  O Paradigma O-O é Baseado no Código e em Dados Encapsulados em uma Única Unidade - o Objeto.
- Elementos Associados a um Objeto
  - Um Conjunto de Variáveis

    Contêm os Dados para o Objeto. O Valor de Cada Variável é Também um Objeto.



## ORIENTAÇÃO A OBJETOS

- Elementos Associados a um Objeto (Cont.)
  - Um Conjunto de Mensagens

    Às quais um Objeto Responde.
  - Um Conjunto de Métodos

    Cada Método é um Corpo de Código para Implementar uma Mensagem. Em geral Retorna um Valor como Resposta a uma Mensagem.



## ORIENTAÇÃO A OBJETOS

- Considerações
  - O termo Mensagem não implica o uso de uma Mensagem Física em uma Rede de Computadores
  - É possível Modificar a Definição de Métodos e Variáveis sem Afetar Outros Objetos
  - A habilidade para Modificar a Definição de um Objeto Sem Afetar o Resto do Sistema é Considerada como uma das Maiores Vantagens do Paradigma O-O



## ORIENTAÇÃO A OBJETOS

- Hierarquia de Classes
  - Classe

    Define e Agrupa Objetos Similares, os quais são chamados de Instâncias da Classe
    - ✓ Um Conjunto de Variáveis cujo Valor é o Conjunto de Todos os Objetos que são Instâncias da Classe



## ORIENTAÇÃO A OBJETOS

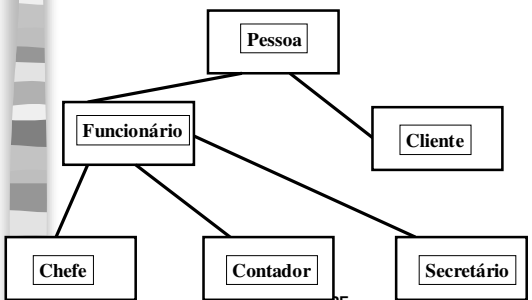
- ✓ Implementação de um Método para a Mensagem ***novo*** que Cria uma Nova Instância da Classe
- Hierarquia de Especialização

  Permite a Representação Direta de Semelhança entre Classes, Colocando as Características Comuns em uma Classe Chamada ***Superclasse*** e as Características Específicas Ficando nas ***Subclasses***.



## ORIENTAÇÃO A OBJETOS

- Um Exemplo no Sistema Bancário



## ORIENTAÇÃO A OBJETOS

- Na Hierarquia de Pessoa:
  - Pessoa é a Superclasse de Funcionário e Cliente
  - Chefe, Contador e Secretário são Subclasses de Funcionário
- Herança de Propriedades

  Variáveis e Métodos são Herdados da Superclasse por suas Subclasses



## ORIENTAÇÃO A OBJETOS

- Generalização e Especialização

  Na Generalização é Obrigatório que Todas as Instâncias da superclasse Estejam Divididas entre as Subclasses
- Herança Múltipla

  Capacidade de Classes em Herdar Variáveis e Métodos de Mais de uma Superclasse.



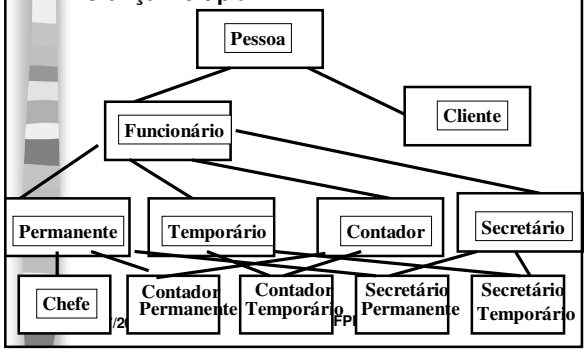
## ORIENTAÇÃO A OBJETOS

- Ambiguidade
  - ✓Incluir os Elementos Conflitantes, Renomeando-os
  - ✓Escolher Herdar de Uma ou de Outra Classe, Baseando-se na Ordem em que as Classes foram Criadas
  - ✓Forçar o Usuário a Fazer uma Escolha no Momento em que a Subclasse for Definida



## ORIENTAÇÃO A OBJETOS

- **Herança Múltipla**






## ORIENTAÇÃO A OBJETOS

- Identidade de Objetos

  Capacidade de Identificar uma Entidade, Mesmo que Todos os Valores de suas Variáveis Mudem com o Tempo
- Formas de Identificação
  - Valor

    Usada em Sistemas Relacionais



## ORIENTAÇÃO A OBJETOS

- Formas de Identificação (Cont.)
  - Nome

    Usada para Variáveis de Programas
  - Embutida

    Nenhuma Identidade Fornecida pelo Usuário é Requerida

## ORIENTAÇÃO A OBJETOS

- Consultas Orientadas a Objeto
  - Pelo Paradigma, Toda Interação com Objetos Deve ser Feita Através do Envio de Mensagens
  - Não Existe um Padrão
    - ✓Trabalha-se numa Versão de SQL, a SQL3
- Modificação de Esquema
  - Mudanças Complexas
  - Freqüentes



## ORIENTAÇÃO A OBJETOS

- Olhar e Modelar o Mundo Real como uma Coleção de Unidades Distinguíveis (Cooperantes e Inter-relacionadas): Objetos
- Abstração e Autonomia
  - Objetos têm Estado e Comportamento
  - Encapsulamento ("Information Hiding")
  - Associação com Outros Objetos
  - Demanda Serviços a Outros Objetos



## ORIENTAÇÃO A OBJETOS

- Classificação
  - Descrição Comum (intenção): Classe
  - Coleção de Objetos Similares (extensão; instâncias)
- Taxonomia
  - Super/Subclasses
  - Herança de Propriedades



## ORIENTAÇÃO A OBJETOS

- Uma Técnica Analítica e Construtiva:
  - Análise de Sistema Orientado a Objetos
  - Projeto de Sistema Orientado a Objetos
  - Construção de Sistema Orientado a Objetos:
    - ✓Programação
    - ✓Interface com o Usuário
    - ✓Banco de Dados



## ORIENTAÇÃO A OBJETOS

- Vantagens da Modelagem Orientada a Objetos
  - Naturalidade
    - ✓Abstração Significativa / Alta Modularidade
    - ✓Separação de Interface / Implementação
    - ✓Domínio da Complexidade
  - Projeto de Sistema Evolutivo
    - ✓Programação Incremental
    - ✓Reusabilidade



## ORIENTAÇÃO A OBJETOS

- Modelos de Dados Orientados a Objetos
  - Propriedades Básicas:
    - ✓Banco de Dados: Coleção de Objetos que são Instâncias de Classes (Definidas no Esquema do BD)
    - ✓Objetos: Valor + Operações
    - ✓Identificação de Objetos
    - ✓Encapsulamento
    - ✓Valores Complexos, Estrutura de Objetos, Objetos Compostos

## ORIENTAÇÃO A OBJETOS

- ✓Hierarquia de Classes (Herança)
  - Overloading
  - Overriding
  - Late Binding
- ➢ Propriedades Adicionais:
  - ✓Relacionamentos Explícitos
  - ✓Tipos Definidos pelo Usuário
  - ✓Versões
  - ✓Modificação Dinâmica do Esquema
  - ✓Regras (Dedutivos, ativos,...)



## ORIENTAÇÃO A OBJETOS

- Requisitos para SGBDOO
  - ➢Fazer os Objetos Criados Persistentes
  - ➢Incompatibilidades entre Linguagens de Programação e de BD devem ser Evitadas: Necessidade de um Modelo de Dados Uniforme nos dois Lados
  - ➢SGBDOOs não devem ser Fortemente Acoplados a Nenhuma Linguagem de Programação



## ORIENTAÇÃO A OBJETOS

- Requisitos para SGBDOO (Cont.)
  - ➢ Deve Gerenciar e Processar Objetos com Estados Estruturados Arbitrariamente, como Também Vários Tipos de Associação Entre Objetos
  - ➢ O Modelo de Dados deve ser Extensível de Modo a poder Acomodar Novos e Pouco Comuns Tipos de Dados
  - ➢ Todas as Propriedades Funcionais de SGBDS devem Estar Disponíveis e com Performance Alta



## OODBMS Manifesto

- Autores:

  M. Atkinson - University of Glasgow
  F.Bancilhon - Altair
  D. De Witt - University of Wisconsin
  K. Dittrich - University of Zurick
  D. Maier - Oregon Graduate Center
  S. Zdonik - Brown University



## OODBMS Manifesto

- Situação
  - ➢Falta de um Modelo de Dados Comum
  - ➢Falta de uma Base Formal (Conceitos Imprecisos)
  - ➢Forte Atividade Experimental (Grande Número de Protótipos)



## OODBMS Manifesto

- Características de Sistemas de Banco de Dados Orientados a Objetos
  - ➢Obrigatórias

    O Sistema Deve Apresentá-las para ser Chamado O-O
  - ➢Opcionais

    Podem ser Adicionadas para Melhorar o Sistema
  - ➢Em Aberto

    Alguns Pontos que o Projetista Pode Escolher para Facilitar sua Implementação

## OODBMS Manifesto

- SGBDOO
  - SGBD
    - ✓Persistência
    - ✓Gerência de Memória Secundária
    - ✓Concorrência
    - ✓Reconstrução
    - ✓Facilidade de consultas “Ad Hoc”



## OODBMS Manifesto

- SOO
  - ✓Objetos Complexos
  - ✓Identidade de Objetos
  - ✓Encapsulamento
  - ✓Tipos ou Classes
  - ✓Herança
  - ✓Overriding
  - ✓Overloading
  - ✓Late Binding
  - ✓Extensibilidade
  - ✓Completude Computacional



## SGBDS Orientados a Objetos

- Características Obrigatórias
  - Objetos Complexos
    - ✓Construídos a Partir de Tipos mais Simples Através de Construtores de Objetos
    - ✓Construtores
      - Mínimo: Conjunto, Lista e Tupla
      - Ortogonais



## SGBDS Orientados a Objetos

- Objetos Complexos (Cont.)
  - ✓Operadores Sobre Objetos
    - Tratar o Objeto como um Todo
    - Propagar Transitivamente para Todos os Seus Componentes



## SGBDS Orientados a Objetos

- Identidade de Objetos

  Uma Objeto tem Existência, Independente de Seu Valor
  - ✓Noções de Equivalência
    - Identidade: Se São o Mesmo Objeto
    - Igualdade: Se Têm o Mesmo Valor
  - ✓Vantagens
    - Compartilhar Componentes
    - Atualização



## SGBDS Orientados a Objetos

- Encapsulamento
  - ✓Necessidade de Distinguir Claramente entre Especificação e Implementação de uma Operação
  - ✓Necessidade de Modularidade
    - Estruturar Aplicações Complexas
    - Ferramenta para Proteção e Autorização

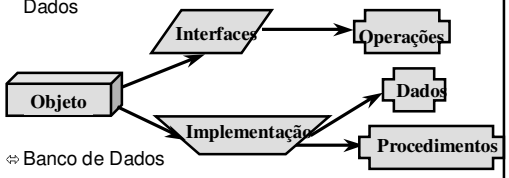
## SGBDS Orientados a Objetos

- Encapsulamento (Cont.)
  - Provê uma Forma de Independência Lógica dos Dados
  - ✓Muda a Implementação de um Tipo, Sem Mudar o Programa que o Utiliza

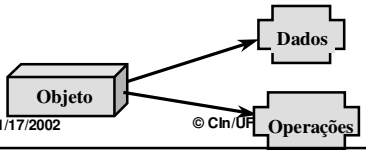


## SGBDS Orientados a Objetos

- Linguagens de Programação: Tipos Abstratos de Dados



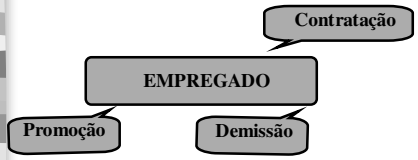

- Banco de Dados






## SGBDS Orientados a Objetos

**Exemplo em BD:**






## SGBDS Orientados a Objetos

- Tipos e Classes
  - ✓Tipo:
    - Define as Características Comuns de um Conjunto de Objetos
    - Tipo Abstrato de Dado
    - Em Linguagens de Programação
      - Aumenta a Produtividade do Programador Garantindo a Correção do Programa



## SGBDS Orientados a Objetos

- Em Linguagens de Programação (Cont.)
  - 80% dos Erros São de Tipos
  - Verificação em Tempo de Compilação



## SGBDS Orientados a Objetos

- ✓Classe:
  - Mesma Especificação de um Tipo
  - Noção em Tempo de Execução
  - Dois Aspectos
    - Fábrica de Objetos (***new***)
    - Depósito de Objetos (Instâncias)

## SGBDS Orientados a Objetos

- Tipos e Classes (Cont.)
  - ✓Noções Bastante Similares
  - ✓O Sistema deve Oferecer algum Mecanismo de Estruturação de Dados, não Necessariamente os dois
  - ✓A Escolha é do Projetista
  - ✓A Noção de esquema de BD é Substituída por Conjunto de Classes ou de Tipos



## SGBDS Orientados a Objetos

- Herança

  Mecanismo Associado a Hierarquias de Tipos ou Classes
  - ✓Vantagens:
    - Ferramenta Poderosa para Modelagem
    - Fatora Especificações e Implementações
    - Facilita a Reusabilidade de Código



## SGBDS Orientados a Objetos

- Herança (Cont.)
  - ✓Tipos de Mecanismos
    - Substituição
    - Inclusão: Noção de Classificação
    - Restrição
    - Especialização



## SGBDS Orientados a Objetos

- Overriding (Sobreposição), Overloading (Sobrecarga), Late Binding (Amarração Postergada)
  - ✓Necessidade

    Diferentes Objetos, Mesmo Nome para Diferentes Operações
  - ✓Exemplo
    - Operação: Exibir
    - Tipos de Objetos: Pessoa, Figura, Grafo



## SGBDS Orientados a Objetos

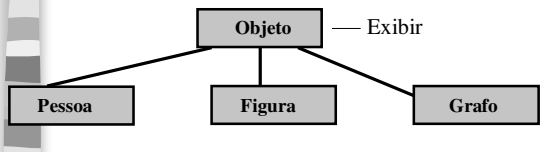
```
For x IN X DO
    BEGIN
        CASE OF TYPE (x)
            PESSOA: EXIBIR_PESSOA(x)
            FIGURA: EXIBIR_FIGURA(x)
            GRAFO:  EXIBIR_GRAFO(x)
        END
    END
```

**Solução Convencional**



## SGBDS Orientados a Objetos

**Solução O-O:**



## SGBDS Orientados a Objetos

- A Operação Exibir é Redefinida para Cada um dos Tipos Envolvidos (Overriding)
- Um Nome Denotando Três Programas Diferentes (Overloading)
- Nomes de Operações devem ser Amarrados a Programas em Tempo de Execução (Late Binding)

  For x IN X Exibir(x)



## SGBDS Orientados a Objetos

- Completude Computacional
  - ✓A Linguagem de Manipulação do BD deve Permitir Expressar Qualquer Função Computável
  - ✓Pode Ser Introduzida Através de Conexão com Linguagens de Programação Existentes



## SGBDS Orientados a Objetos

- Extensibilidade
  - ✓Conjunto de Tipos deve ser Extensível
    - Permitir a Definição de Novos Tipos
    - Sem Distinção entre Tipos Pré-definidos e Definidos pelo Usuário
    - Não é Necessário que a Coleção de Construtores de Tipo seja Extensível



## SGBDS Orientados a Objetos

- Persistência

  Habilidade de dados Sobreviverem à Execução de um Processo
  - ✓É Ortogonal: Cada Objeto, Independente de seu Tipo, pode Tornar-se Persistente
  - ✓Deve ser Implícita



## SGBDS Orientados a Objetos

- Gerenciamento de Memória Secundária
  - ✓Gerenciamento de Índices
  - ✓Segmentação / Paginação de Dados
  - ✓Seleção de Caminhos de Acesso
  - ✓Otimização de Consultas

**Performance**



## SGBDS Orientados a Objetos

- Concorrência

  Garantir uma Coexistência Harmoniosa entre Usuários Trabalhando Simultaneamente no BD
- Recuperação

  No Caso de Falhas, Trazer o Sistema para Algum Estado Coerente

## SGBDS Orientados a Objetos

- Facilidade de Consultas Ad-Hoc
  - ✓Ser de Alto Nível (Declarativa)
  - ✓Ser Eficiente (Otimizável)
  - ✓Independente de Aplicação



## SGBDS Orientados a Objetos

- Alguns Pontos sem Consenso
  - ✓Definição de Visões e Dados Derivados
  - ✓Utilitários de Administração de BD
  - ✓Restrições de Integridade
  - ✓Facilidade de Evolução de Esquema de BD



## SGBDS Orientados a Objetos

- Características Opcionais
  - Herança Múltipla
  - Verificação de Tipos e Inferência de Tipos
  - Distribuição
  - Transações de Projeto
  - Versões



## SGBDS Orientados a Objetos

- Características em Aberto
  - Paradigma de Programação
  - Sistema de Representação
  - Sistema de Tipos
  - Uniformidade
- Classificação
  - Estruturalmente O-O: Objetos Complexos
  - Comportalmente O-O: Extensibilidade
  - Totalmente O-O: Ambos