

# Banco de Dados

- Banco de Dados;
- Sistema de Gerenciamento de Banco de Dados;
- Modelo Entidade-Relacionamento;
- Banco de Dados Relacional;
- Banco de Dados Distribuído;
- Banco de Dados Federado;
- Projeto de Banco de Dados.

# Banco de Dados

- “Banco de dados é uma coleção de unidades de dados físicos que são relacionados entre si de uma maneira específica” (GUIDE-SHARE, 1970);
- “Uma banco de dados consiste de todas as ocorrências de registros controlados por um esquema específico” (CODASYL, 1971);
- “Um banco de dados é uma coleção de dados, organizados e integrados, que constituem uma representação natural de dados, e que possa ser utilizada por todas as aplicações relevantes sem duplicação de dados” (PALMER, 1975).

# Sistema de Processamento de Arquivos

- Antes da concepção dos BDs o registro das informações eram feitos através de arquivos.
- Desvantagens:
  - Redundância e Inconsistência de dados;
  - Dificuldade no acesso aos dados;
  - Isolamento de dados;
  - Anomalias de acesso concorrente;
  - Problemas de segurança;
  - Problemas de privacidade;
  - Dificuldades de manutenção;
  - Problemas de integridade.

# Sistema de Gerenciamento de Banco de Dados (SGBD)

- Tem como objetivo prover um ambiente que seja adequado e eficiente para recuperar, armazenar e manter as informações pertencentes a um banco de dados;
- Justificativa: Necessidade de haver a separação entre os problemas específicos de aplicações e problemas de armazenamento de dados, ou seja, independência entre dados e programas;
- São projetados para gerenciar grandes grupos de informações;
- As primeiras implementações de SGBDs datam do final da década de 60.

# Características dos SGBDs

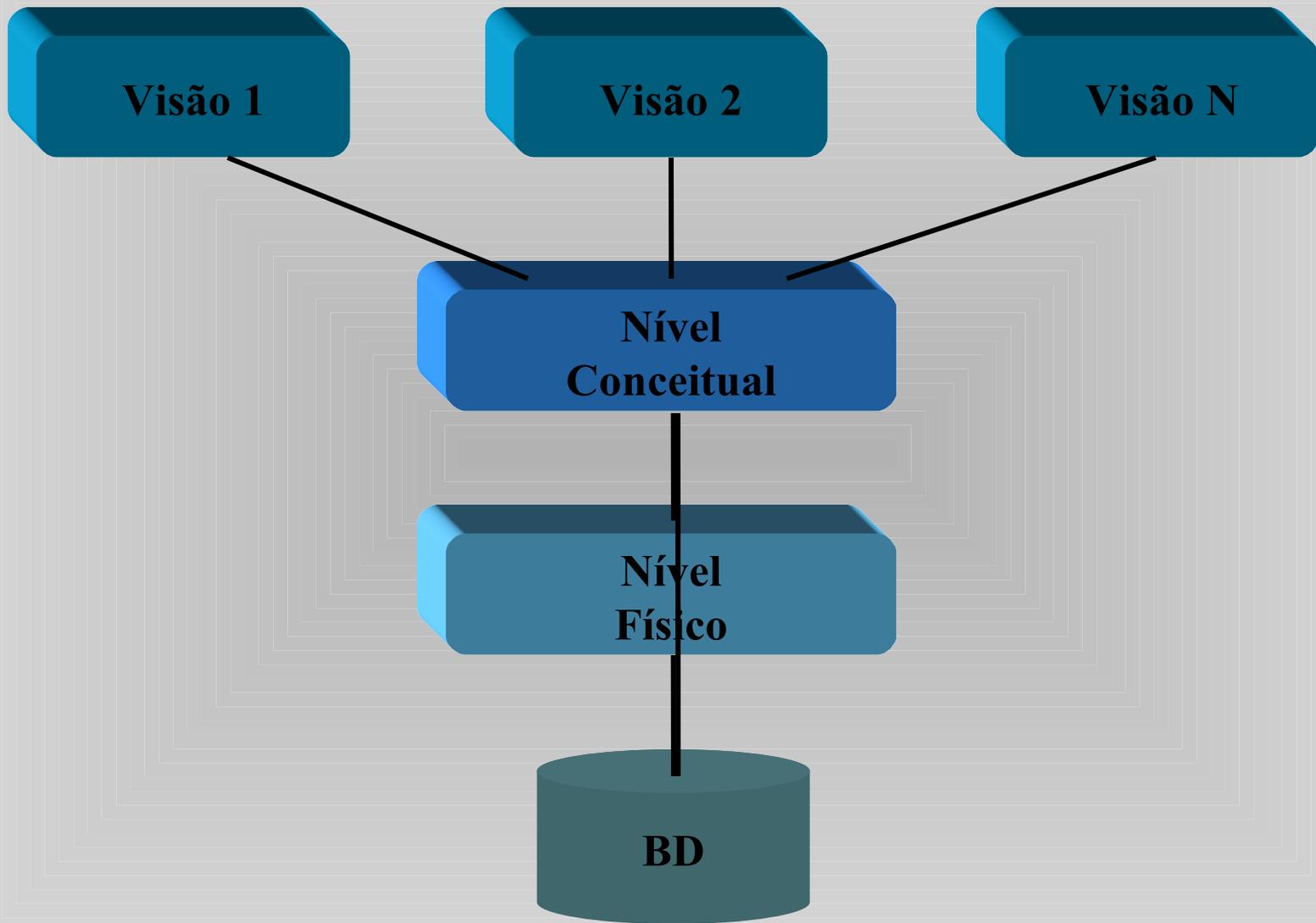
- Eles devem:
  - Definir as estruturas de armazenamento dos dados;
  - Definir os mecanismos para manipulação dos dados;
  - Garantir a segurança dos dados contra ações externas;
  - Garantir a integridade dos dados armazenados;
  - Controlar o acesso concorrente aos dados;
  - Permitir a independência dos dados armazenados;
  - Permitir e manter o relacionamento entre os dados;
  - Controlar a redundância dos dados;
  - Garantir um bom desempenho mesmo com uma grande massa de dados.

# Abstração de Dados

- O grande objetivo de um sistema de banco de dados é prover aos usuários uma visão abstrata dos dados;
- O sistema omite certos detalhes de como os dados são armazenados e mantidos, uma vez que muitos dos seus usuários não são especialistas;
- A complexidade está escondida deles através de diversos níveis de abstração;
- As arquiteturas de bancos de dados podem possuir 1, 2 ou 3 níveis de abstração.

# Níveis de Abstração

- **Nível Físico:** É o nível mais baixo de abstração e descreve como os dados estão realmente armazenados;
- **Nível Conceitual:** Descreve todos os dados que estão armazenados de fato no banco de dados e as relações existentes entre eles;
- **Nível de Visões:** É o mais alto nível de abstração e descreve o banco de dados em partes que são de interesse de cada usuário / aplicação.



# Modelos de Dados

- Estes conceitos fundamentam as estruturas nas quais um banco de dados pode ser construído;
- São utilizadas para especificar a descrição, os relacionamentos e a semântica dos dados e suas restrições de consistência;
- Existem 3 modelos diferentes: 2 para o nível lógico e 1 para o nível físico.

# Tipos de Modelos de Dados

- **Modelo Lógico Baseado em Objetos:** Se caracteriza por tratar os dados com sendo objetos. Ex: Entidade-Relacionamento, Orientado a Objetos, Binário;
- **Modelo Lógico Baseado em Registros:** Neste modelo o banco de dados é estruturado em registros de formato fixo e tipos definidos. Ex: Relacional, Rede, Hierárquico;
- **Modelo Físico de Dados:** Descreve os dados no nível mais baixo do modelo de abstração. Ex: unificador (uniting model), Estrutura de Memória (frame memory).

# Conceitos Relacionados

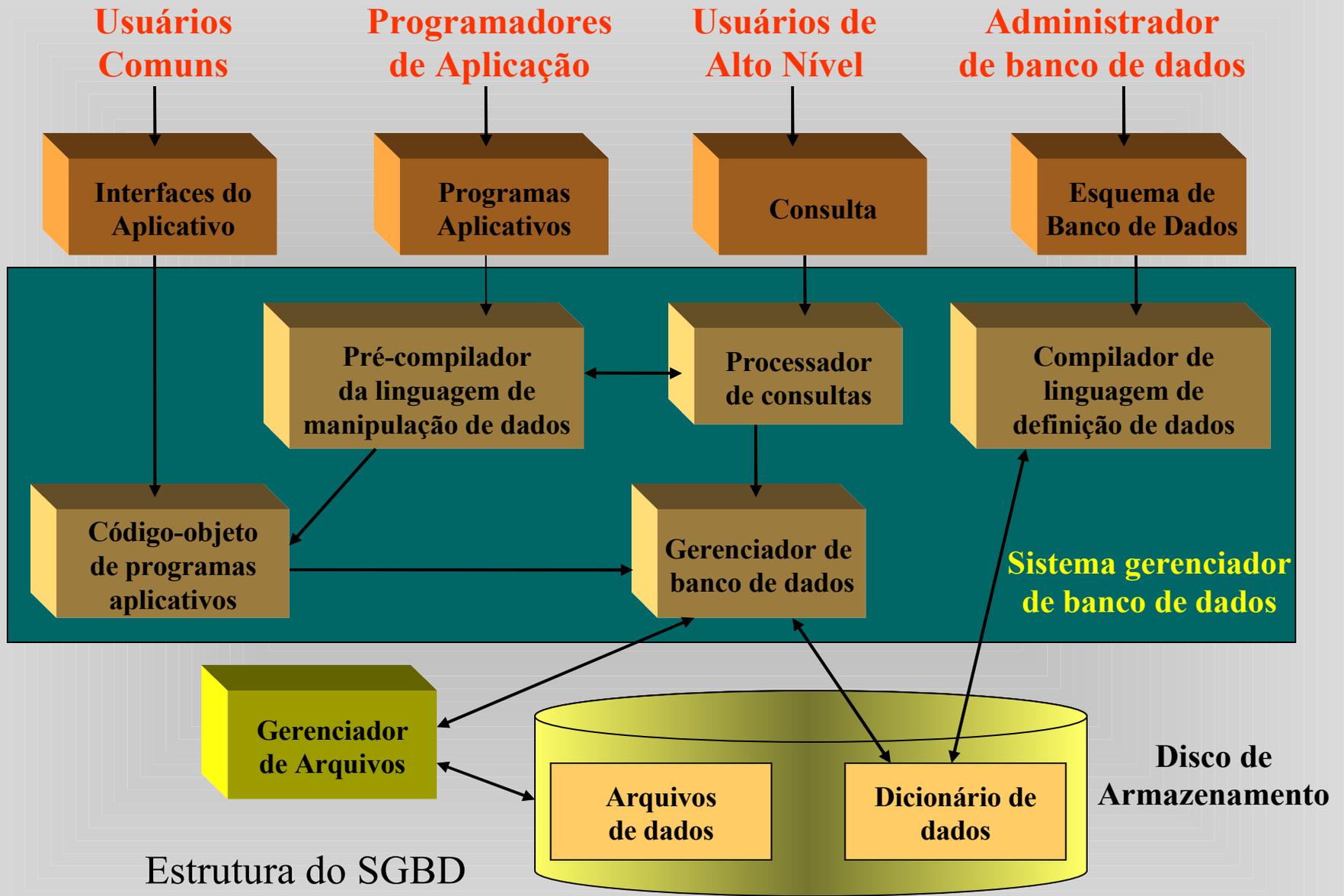
- **Esquema:** É chamado de esquema todo o projeto lógico e físico de um banco de dados;
- **Linguagem de Definição de Dados (DDL):** É a linguagem utilizada para descrever o esquema do banco de dados;
- **Linguagem de Manipulação de Dados (DML):** É a linguagem que permite aos usuários ter acesso e manipular os dados do banco de dados.

# Independência de Dados

- Independência de dados é a capacidade de modificar a definição de um esquema em um nível de abstração sem afetar a definição dos outros esquema;
- Níveis de independência:
  - **Independência física de dados:** É a habilidade de se modificar o esquema físico sem a necessidade de rescrever o modelo conceitual.
  - **Independência lógica de dados:** É a habilidade de se modificar o esquema conceitual sem a necessidade de rescrever as aplicações.

# Estrutura Geral de um SGBD

- Um sistema de banco de dados é dividido em módulos que tratam de cada uma das responsabilidades do sistema como um todo;
- O sistema operacional do computador fornece apenas os serviços mais básicos e o sistema de banco de dados precisa ser construído sobre essa base.



# Módulos de um SGBD

- **Gerenciador de Arquivos:** Gerencia a alocação do espaço na armazenagem em disco e as estruturas de dados usadas para representar a informação;
- **Gerenciador do Banco de Dados:** Fornece a interface entre os dados de baixo nível armazenados em disco e os programas aplicativos;
- **Processador de Consultas:** Traduz comandos numa linguagem de consulta em instruções de baixo nível que o gerenciador do banco de dados pode interpretar.

# Módulos de um SGBD (cont.)

- **Pré-compilador da DML:** Converte comandos da DML embutidos em um aplicativo para chamadas de procedimento normal na linguagem hospedeira.
- **Compilador da DDL:** Converte comandos DDL em um conjunto de tabelas contendo metadados (dados sobre dados). Estas tabelas fazem parte do Dicionário de Dados.

# Estruturas Utilizadas

- **Arquivos de dados:** Armazenam os dados do banco de dados.
- **Dicionário de dados:** Armazena metadados sobre a estrutura do banco de dados.
- **Índices:** Fornecem acesso rápido aos itens de dados guardando determinados valores.

# Usuários do SGBD

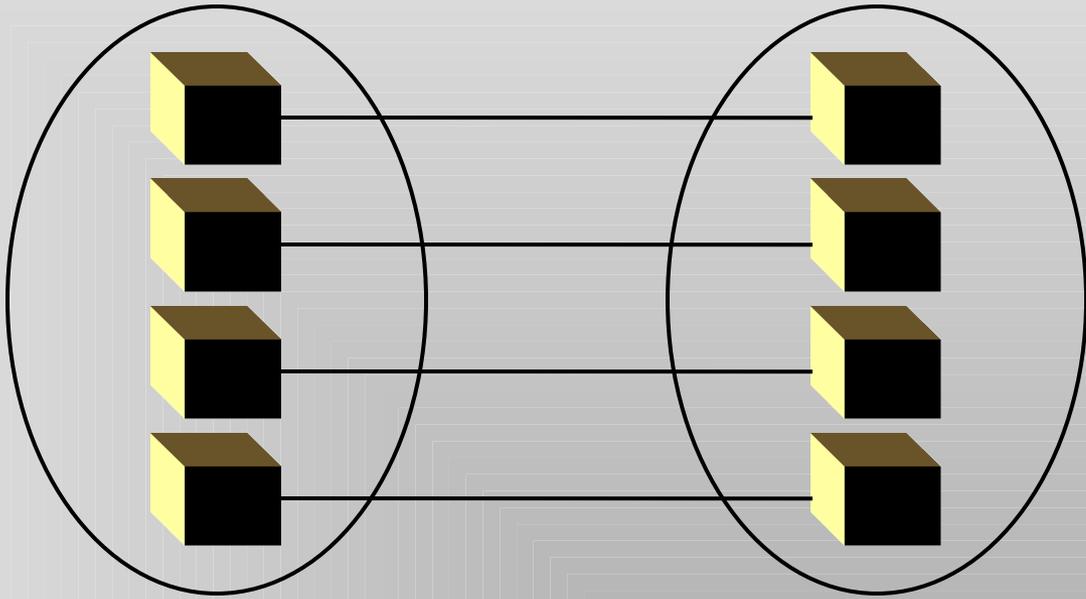
- **Administrador:** Tem o controle central dos dados e programas de acesso;
- **Programadores de Aplicação:** São profissionais que interagem com o sistema por meio de chamadas DML embutidas em programas;
- **Usuários de Alto Nível:** Formulam consultas em uma linguagem de consulta e cada uma delas é submetida ao processador de consulta que gera comandos DML;
- **Usuários Especializados:** São usuários que escrevem aplicativos não tradicionais. Ex: Sistemas especialistas, sistemas que armazenam dados com tipos complexos;
- **Usuários Comuns:** Esses usuários interagem com o sistema invocando aplicativos previamente escritos.

# Modelo Entidade-Relacionamento

- É baseado na percepção do mundo real que consiste em um conjunto de objetos chamados **entidades** e nos **relacionamentos** entre esses objetos;
- **Entidade**: Uma entidade é um objeto que existe e é distinguível dos outros objetos;
- **Relacionamento**: Um relacionamento é uma associação entre diversas entidades. A maioria dos relacionamentos em um banco de dados é binária, isto é, envolve dois conjuntos de entidades. Entretanto podem existir conjuntos de relacionamentos que envolvam mais de dois conjuntos de entidades.

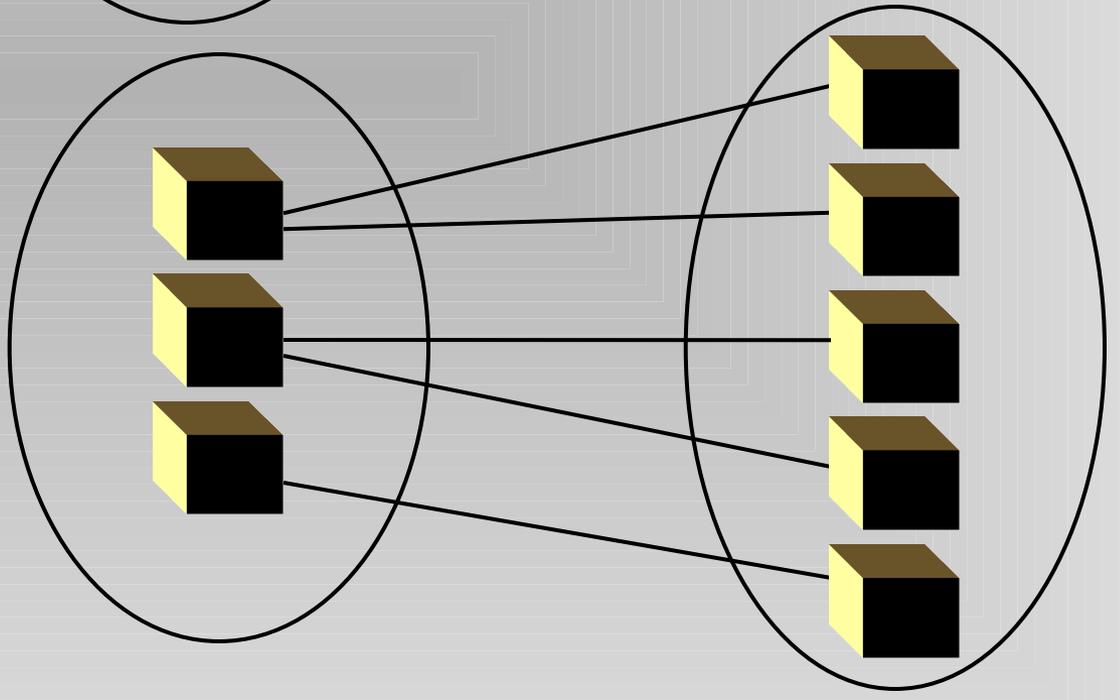
# Cardinalidade dos Relacionamentos

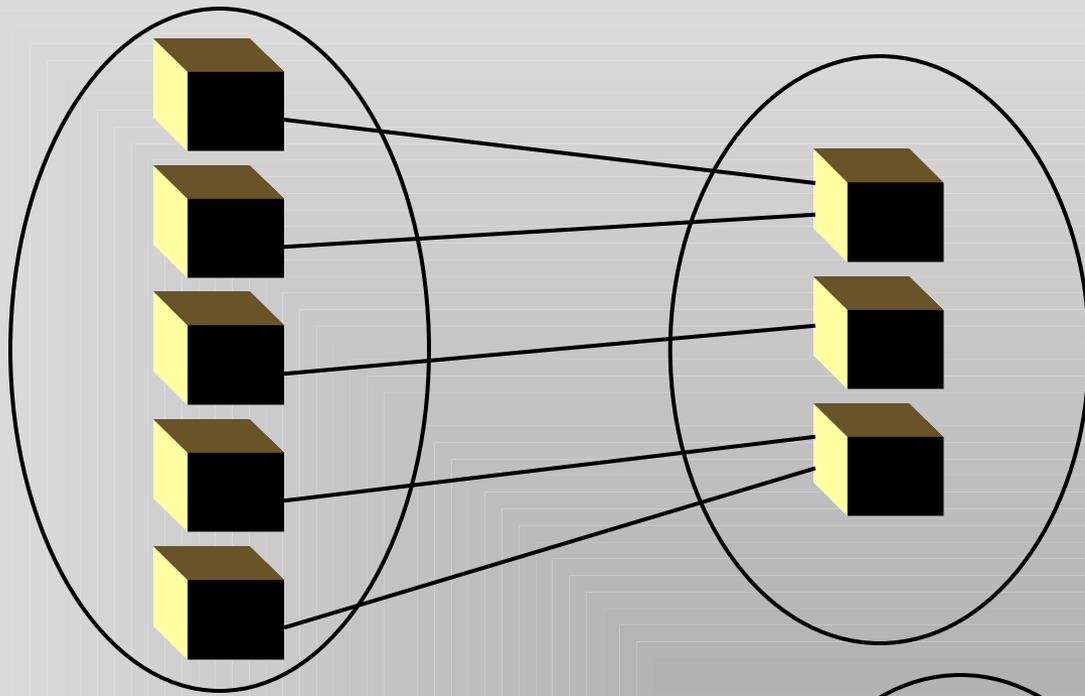
- **Um-para-um**: Uma entidade de A está associada no máximo a uma entidade de B, e vice-versa;
- **Um-para-muitos**: Uma entidade de A está associada a qualquer número de entidades de B. Uma entidade de B, porém, pode estar associada no máximo a uma entidade de A;
- **Muitos-para-um**: Uma entidade de A está associada no máximo a uma entidade de B. Uma entidade de B, entretanto, pode estar associada a qualquer número de entidades de A;
- **Muitos-para-muitos**: Uma entidade de A está associada a qualquer número de entidades de B, e vice-versa.



**Relacionamento um-para-um**

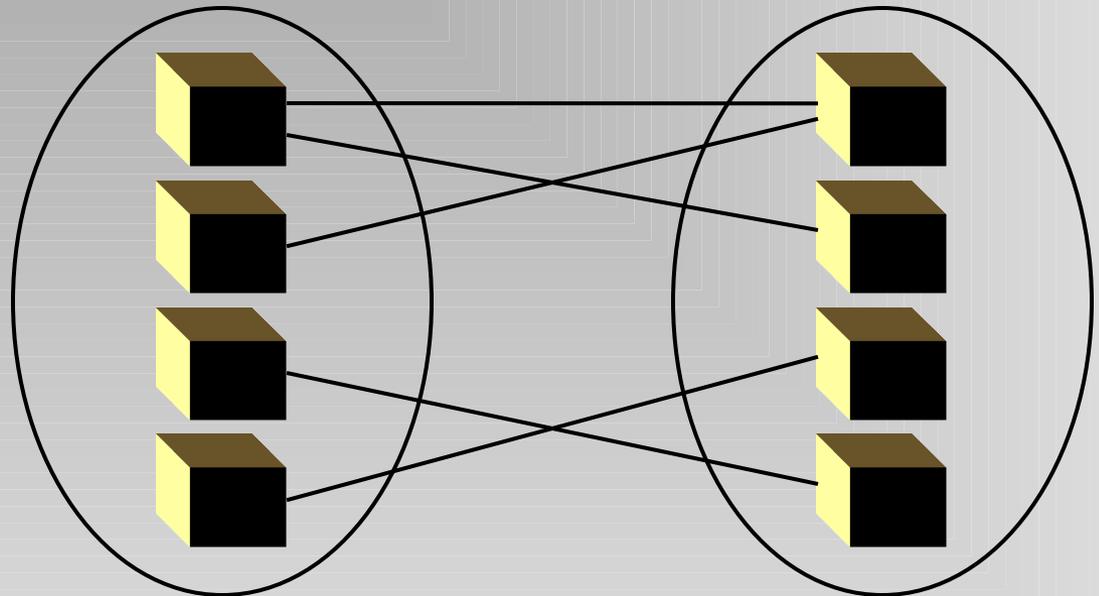
**Relacionamento um-para-muitos**





**Relacionamento muitos-para-um**

**Relacionamento muitos-para-muitos**

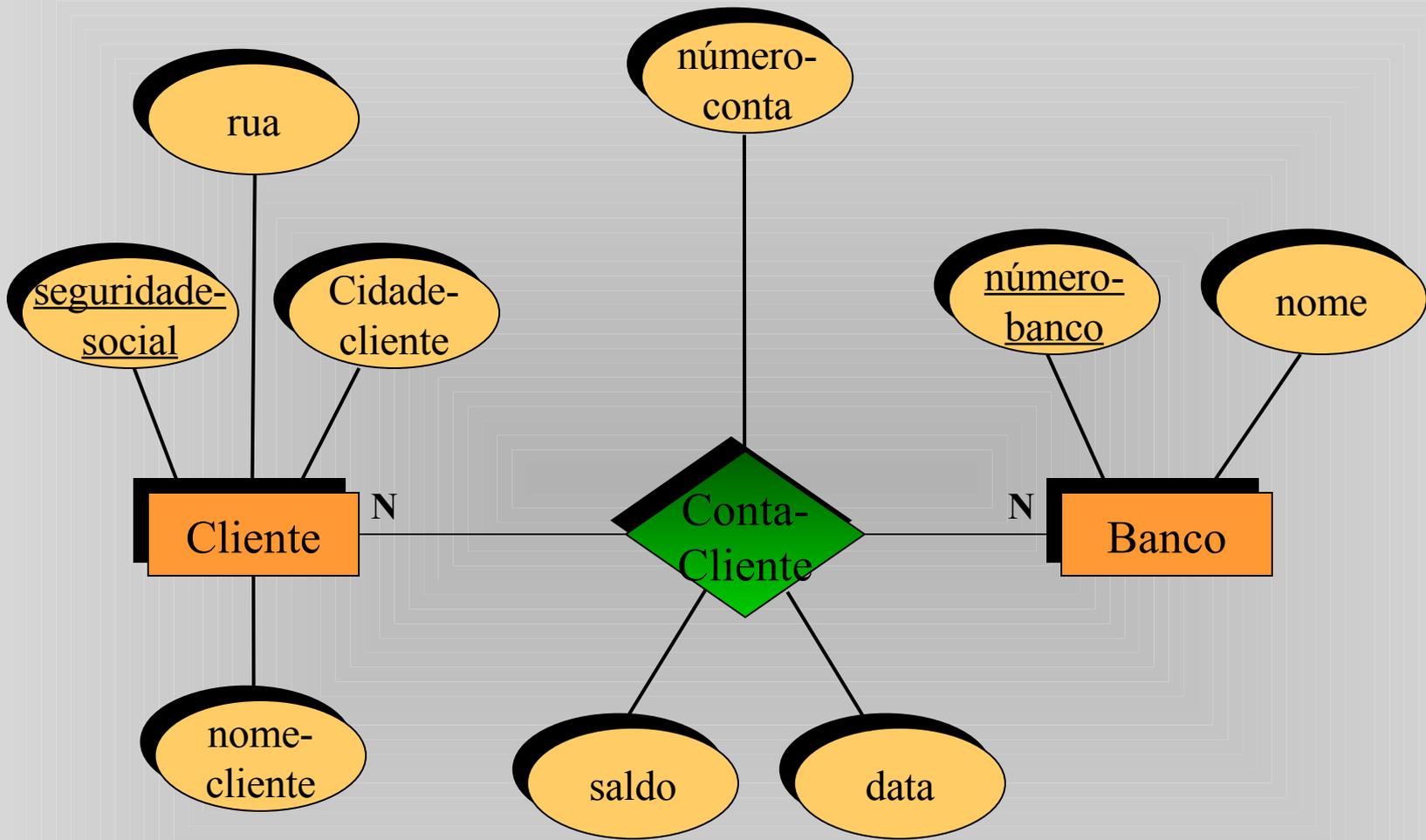


# Chaves

- **Superchave:** É um conjunto de um ou mais atributos que permite identificar univocamente uma entidade no conjunto de entidades;
- **Chave candidata:** É uma superchave que não tenha nenhum subconjunto próprio que seja uma superchave;
- **Chave primária:** É o conjunto de atributos que melhor identifica univocamente uma entidade dentro de um conjunto.

# Diagrama Entidade-Relacionamento

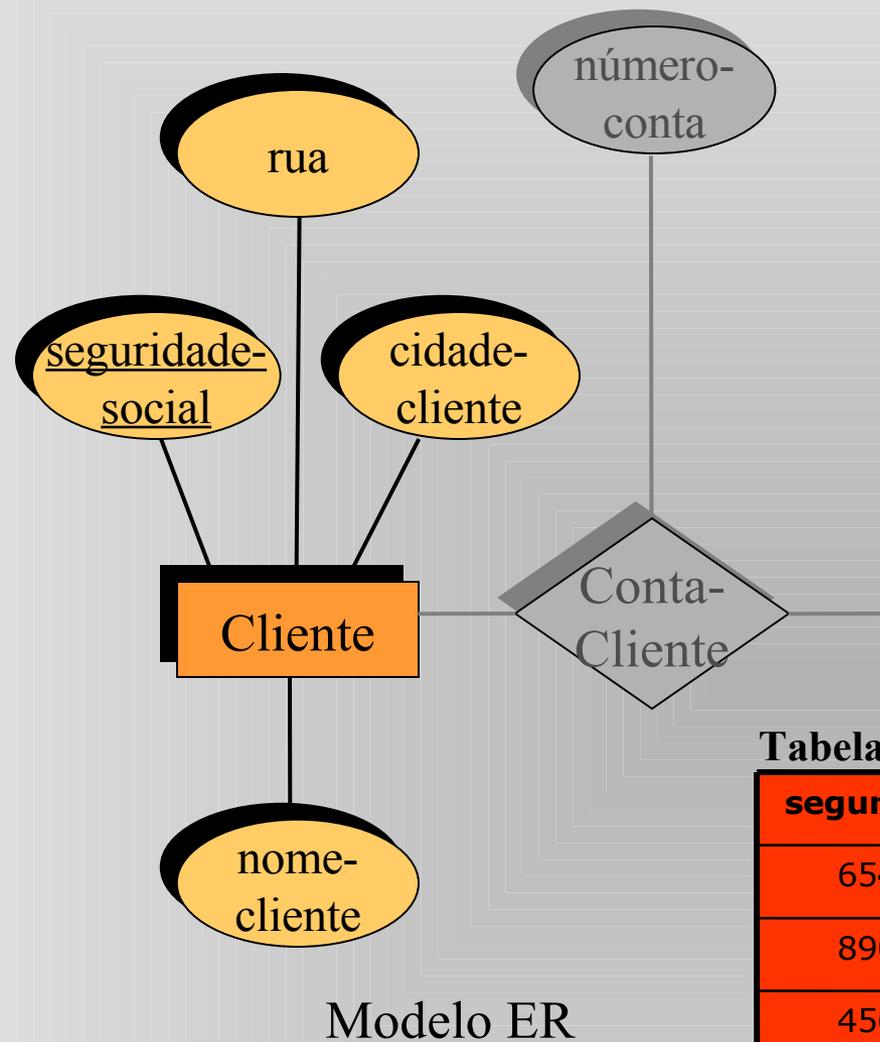
- A estrutura lógica de um banco de dados pode ser expressa graficamente através de um diagrama ER. Tal diagrama consiste nos seguintes componentes:
  - **Retângulos**: Representam conjuntos de entidades;
  - **Elipses**: Representam atributos;
  - **Losangos**: Representam conjuntos de relacionamentos;
  - **Linhas**: Ligam atributos a conjuntos de entidades e conjuntos de entidades a conjuntos de relacionamentos.



Exemplo de um diagrama ER

# Representação ER em tabelas

Para o banco de dados relacional um diagrama ER pode ser representado por uma coleção de tabelas. Para cada conjunto de entidades e relacionamentos do diagrama, existe uma tabela designada com o mesmo nome e que as representa.



## Modelo Lógico Relacional

**Tabela Cliente**

| <b>seguridade-social</b> | <b>nome-cliente</b> | <b>rua</b> | <b>cidade-cliente</b> |
|--------------------------|---------------------|------------|-----------------------|
| 654-32-1098              | Oliver              | Main       | Harrison              |
| 890-12-3456              | Harris              | North      | Rye                   |
| 456-78-9012              | Marsh               | Park       | Stamford              |
| 121-21-2121              | Brill               | Nassau     | Rye                   |

Modelo ER

# Banco de Dados Relacional

- Um banco de dados relacional consiste em uma coleção de tabelas, cada qual designada por um nome único;
- Uma linha numa tabela representa um relacionamento entre um conjunto de valores;
- Uma vez que uma tabela é uma coleção de relacionamentos, existe uma correspondência íntima entre o conceito de tabela e o conceito matemático de relação;
- É a partir desta correlação que o modelo de dados relacional tira seu nome.

# Chave Primária e Estrangeira

- As noções de superchave, chave candidata, chave primária também são aplicáveis ao modelo relacional;
- Podemos estender este conceito incluindo a noção de chave estrangeira que serve como forma de transformar um relacionamento fraco entre entidades em um relacionamento forte.

Tabela Agência

| número-agência | nome-agência |
|----------------|--------------|
| 10             | Downtown     |
| 13             | Mianus       |
| 15             | Redwood      |
| 21             | Pownal       |

## Chave Primária e Estrangeira

- Chave Primária**
- Chave Estrangeira**

Tabela Empréstimo

| número-agência | número-empréstimo | número-cliente | quantia |
|----------------|-------------------|----------------|---------|
| 10             | 17                | Jones          | 1000    |
| 13             | 29                | Curry          | 500     |
| 21             | 29                | Willians       | 1200    |
| 15             | 23                | Smith          | 2000    |

**Chave Estrangeira + Chave Candidata = Chave Primária**

# Restrições de Integridade

- As restrições de integridade fornecem meios para assegurar que mudanças feitas no banco de dados não resultem na perda da consistência sobre estes dados. Existem várias formas de se manter a integridade de um banco de dados:
  - **Declaração de Chaves**: Serve como forma de se garantir a distinção entre as entidades, pois não poderá haver duas entidades com o mesmo valor de chave primária;
  - **Restrição de Domínio**: Cada atributo tem um conjunto possível de valores chamado domínio. O princípio por trás do domínio é garantir a veracidade dos valores;

# Restrições de Integridade (cont.)

- **Integridade Referencial:** É a forma de assegurar que um valor que aparece em uma relação para um dado conjunto de atributos apareça também em um certo conjunto de atributos em outra relação;
- **Dependências Funcionais:** É uma generalização da noção de chave, ou seja, são restrições sobre o conjunto de relações válidas.

# Gerenciamento de Transações

- Uma transação é qualquer execução de um programa e / ou usuário em um SGBD;
- É a unidade básica de mudança percebida pelo SGBD;
- O SGBD deve escalonar de forma segura os acessos concorrentes as dados do BD;
- O usuário deve pensar que sua consulta ou alteração é a única a ser executada isoladamente no DB;
- Para assegurar o controle de concorrência os SGBD implementam um **protocolo de travamento**;
  - Travamento compartilhado;
  - Travamento exclusivo.

# Linguagens de Consulta

- São linguagens que o usuário utiliza para requisitar informações do banco de dados;
- Elas podem ser classificadas em dois tipos:
  - **Procedurais**: O usuário instrui o sistema a executar uma seqüência de operações no banco de dados a fim de computar o resultado desejado. Ex: Álgebra relacional;
  - **Não-procedurais**: O usuário descreve a informação desejada sem fornecer um procedimento específico para obter tal informação. Ex: Cálculo relacional de tupla e cálculo relacional de domínio.

# SQL

- SQL (Structured Query Language);
- Usa uma combinação da álgebra relacional e construções de cálculo relacional;
- Foi desenvolvida pela IBM no início dos anos 70 e mais tarde se tornou um padrão ANSI;
- Se estabeleceu como a linguagem padrão para banco de dados relacional;
- Embora seja chamada de “linguagem de consulta” ela contém outras capacidades além de consultas a banco de dados;
- Inclui recursos para definição de estruturas, modificação e restrições de dados.

# Partes da Linguagem

- **Data definition language (DDL):** A DDL fornece comandos para criação de índices e chaves, definição, remoção e modificação de esquemas de relações (tabelas), etc.;
- **Interactive data manipulation language (DML):** A DML inclui uma linguagem de consulta baseada na álgebra relacional e no cálculo relacional de tuplas. Contém também comandos para inserir, remover e modificar registros;
- **Embedded data manipulation language:** É a forma embutida da SQL que foi projetada para utilização em linguagens de programação de uso geral como Cobol, C, Pascal;

# Partes da Linguagem (cont.)

- **View definition:** É a parte da DDL que inclui comandos para definição de visões;
- **Authorization:** É a parte da DDL que inclui comandos para especificação de autorização de acesso a relações e visões;
- **Integrity:** Fornece mecanismos para a verificação de integridade;
- **Transaction control:** Fornece comandos para especificação de início e fim de transações.

# Comandos Básicos em SQL

- A estrutura básica de uma expressão SQL para consulta consiste em três cláusulas: **Select**, **From** e **Where**.
  - **Select**: Corresponde a operação projeção da álgebra relacional. É usada para listar os atributos desejados no resultado de uma consulta;
  - **From**: Corresponde a operação produto cartesiano da álgebra relacional. Ela lista as relações a serem examinadas na avaliação da expressão;
  - **Where**: Corresponde ao predicado da seleção da álgebra relacional. Consiste em um predicado envolvendo atributos de relações que aparecem na cláusula From.

# Comandos de Seleção

- Projeção: `select idfuncionario, nome_funcionario  
from funcionario`
- Seleção: `select nome_cliente  
from deposito  
where nome_agencia = 'Central'`
- Junção: `select f.nome_funcionario, d.nome_dependente  
from funcionario f, dependente d  
where f.idfuncionario = c.idfuncionario`

# Comandos de Modificação

- Remoção: `delete from deposito  
where nome_cliente = 'Smith'`
- Inserção: `insert into deposito (nome_agencia, numero_conta,  
nome_cliente, saldo)  
values ('Central', '9732', 'Smith', '1200')`
- Atualização: `update deposito  
set saldo = saldo * 1,05  
where saldo > 1000`

# Comandos de Criação

- Criação: 

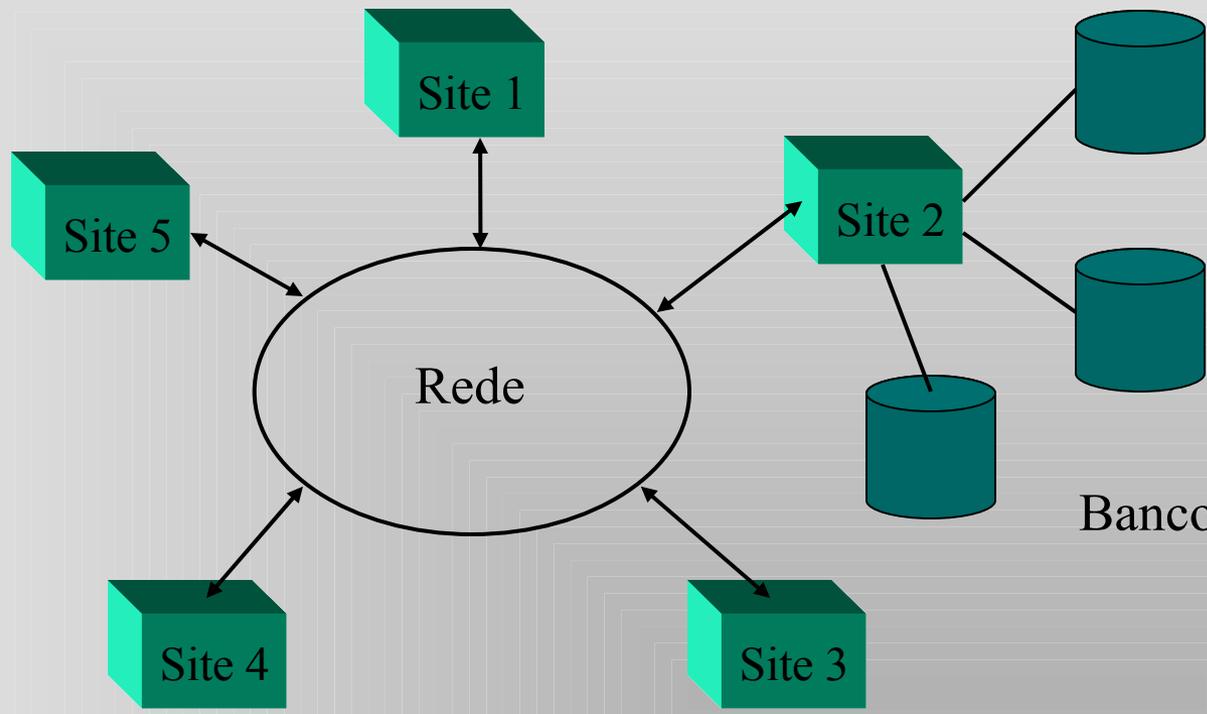
```
create table deposito (  
    nome_agencia string not null,  
    numero_conta integer not  
    null,  
    nome_cliente string,  
    saldo float  
    primary key (nome_agencia));
```
- Alteração: 

```
alter table deposito  
    add saldo  
    float;
```
- Deleção: 

```
drop table deposito;
```

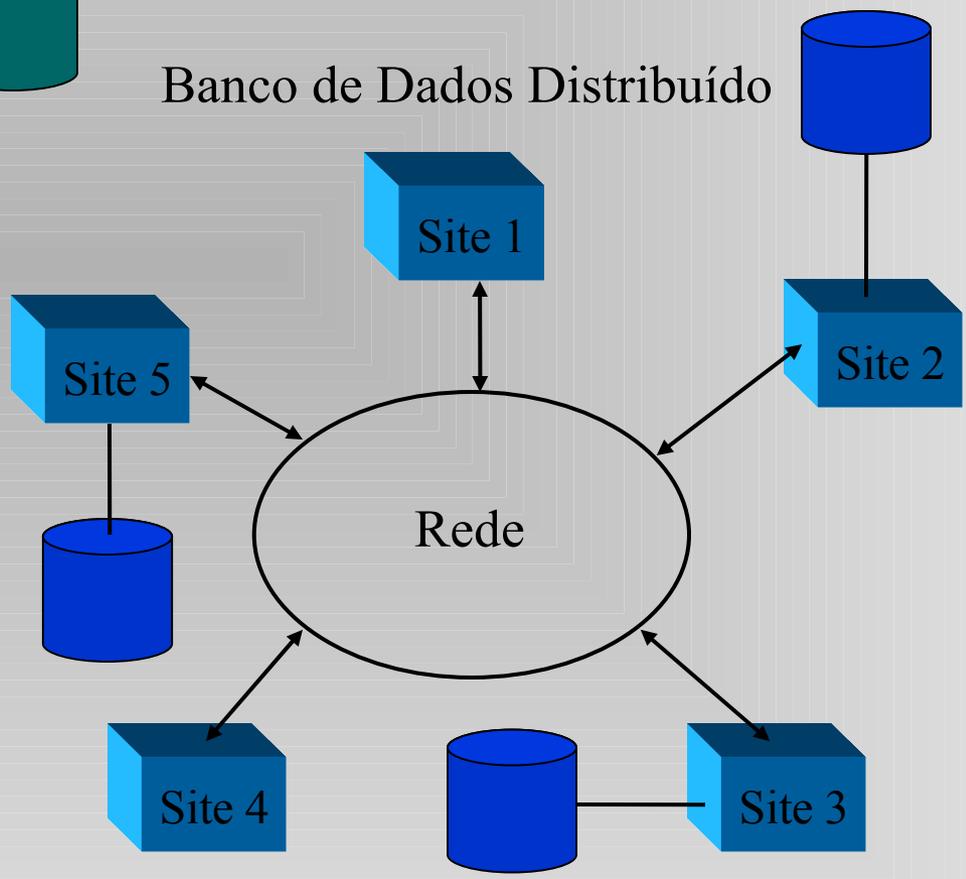
# Banco de Dados Distribuídos

- O banco de dados é armazenado em diversos computadores que se comunicam através de uma tecnologia de rede;
- Um banco de dados distribuído consiste em uma coleção de nós, cada qual podendo participar na execução de transações que fazem acesso a dados em um ou diversos nós;
- A execução das transações entre os nós devem ser transparente para o usuário.



Banco de Dados Centralizado

Banco de Dados Distribuído



# Prós

- Controle distribuído;
- Diminuição nos custos de comunicação e equipamentos;
- Confiabilidade;
- Disponibilidade;
- Facilidade no crescimento modular (escalabilidade);
- Aceleração das consultas.

# Contras

- Maiores custo e complexidade de desenvolvimento, tanto do SGBD quanto do esquema de dados;
- Dificuldades da coordenação e controle dos dados armazenados;
- Aumento na troca de mensagens para sincronização e controle.

# Autonomia

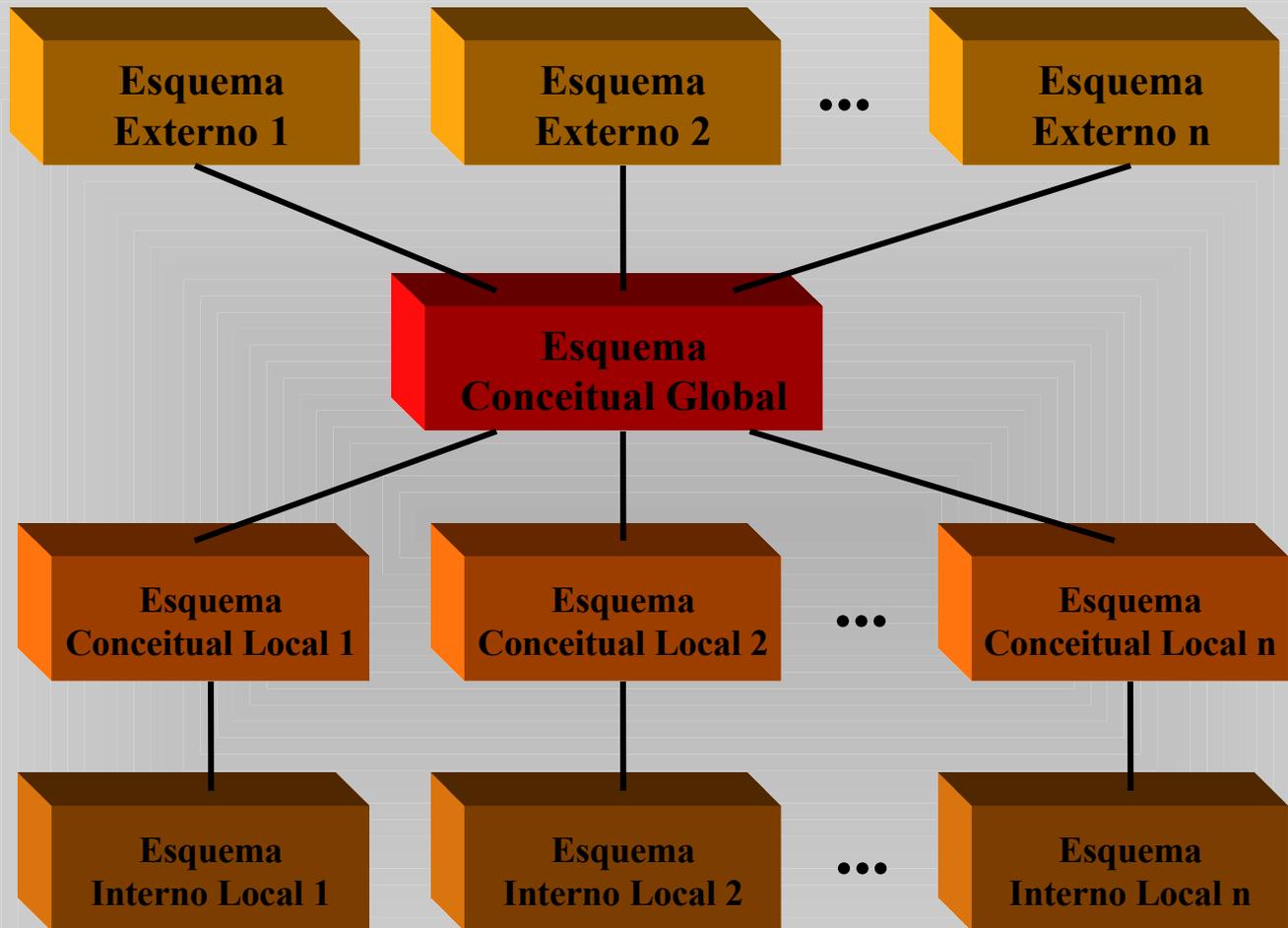
- Autonomia se refere a distribuição do controle;
- Indica o grau de independência de um SGBD;
- Funções consideradas:
  - Se os componentes do sistema trocam informação;
  - Se eles podem independentemente executar transações;
  - Se é permitido que um componente influencie os outros.

# Dimensões de Autonomia

- **Autonomia de Projeto:** SGBDs individuais são livres para usar o modelo de dados e as técnicas de gerenciamento de transações que eles preferirem;
- **Autonomia de Participação:** Cada SGBD local tem o direito de decidir como irá participar no sistema distribuído;
- **Autonomia de Comunicação:** Cada SGBD é livre para decidir que tipo de informação ele deseja trocar e o software que fará o controle das transações;
- **Autonomia de Execução:** Cada SGBD pode executar as transações que são submetidas a ele de várias formas basta ele escolher qual é a melhor.

# Distribuição

- Considera a distribuição física dos dados;
- O usuário vê os dados como um agrupamento lógico, ou seja, para o usuário os dados estão centralizados.
- **Modelo Peer-to-peer:**
  - É o modelo que representa a arquitetura do BD distribuído;
  - Cada máquina tem um SGBD completo e pode se comunicar com outras máquinas para executar consultas e transações;
  - Existe um esquema conceitual global que representa o BD distribuído de forma total.



Arquitetura de referência peer-to-peer

# Heterogenidade

- Heterogenidade pode ocorrer de várias formas em um sistema distribuído;
- Heterogenidades mais comuns:
  - Diferença de Hardware;
  - Diferença de SGDB;
  - Diferença de protocolos de rede;
  - Diferença entre modelos de dados;
  - Diferença de protocolos de gerenciamento de transações.

# Questões relacionadas ao armazenamento distribuído

- **Duplicação:** O sistema mantém diversas cópias idênticas armazenadas em nós diferentes;
  - Vantagens: Maior disponibilidade e aumento do paralelismo;
  - Desvantagem: aumento da sobrecarga na atualização;
- **Fragmentação:** A relação é dividida em vários fragmentos. Cada fragmento é armazenado em um nó diferente. **Existem dois tipos de fragmentação: Horizontal e vertical;**

# Banco de Dados Federados

- São BDs centralizados que recebem uma camada adicional de software para cooperar de forma distribuída;
- Um sistema de banco de dados federado pode ser definido como:
  - Uma coleção de sistemas de banco de dados independentes;
  - Cooperativos;
  - Parcialmente ou totalmente heterogêneos;
  - Autônomos;
  - Que permitem o compartilhamento de todos ou alguns de seus dados.
- Todas essas características não afetam as aplicações locais.

# Consulta Federada

- As principais soluções para os problemas encontrados em consultas a BDs federados estão relacionadas ao uso das seguintes ferramentas:
- **Mediador**: Também chamado de servidor, é um software usado para permitir a interoperabilidade entre dois ou mais SGBDs. O acesso aos dados heterogêneos é efetuado através de consultas que são submetidas ao mediador, que por sua vez as transforma em subconsultas a serem enviadas as SGBDs componentes, em seguida seus resultados são combinados e retornados;

# Consulta Federada (cont.)

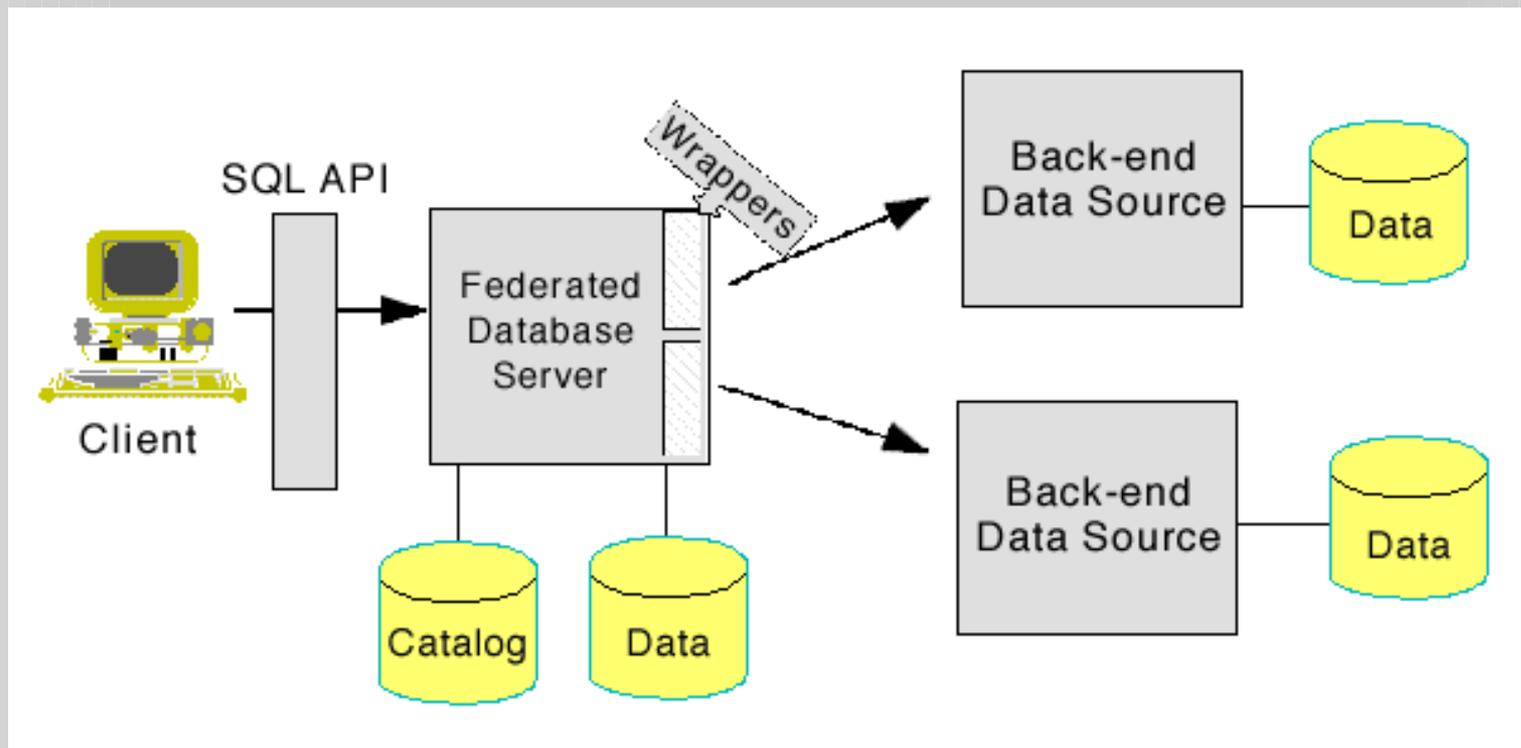
- **tradutores/adaptadores**: Também chamado de *Wrapper*, convertem os dados fonte para um modelo de dados comum, e também convertem consultas de aplicações em consultas específicas das fontes de informação envolvidas na consulta;
- **Visões**: São usadas como mecanismo que auxilia a integração dos componentes de um sistema federado. Servem como interface para o acesso a cada um dos componente.

# Passo da Consulta – Estudo de Caso

- Primeiramente o servidor federado otimiza a consulta desenvolvendo um plano de execução em a mesma é decomposta em fragmentos;
- O servidor submete estes fragmentos de consultas candidatas chamados “pedidos” para os *wrappers*, se o fragmento da consulta é aplicado a fonte de dados;
- Quanto um *wrapper* recebe o pedido, ele determina qual porção do fragmento da consulta pode ser executada pela fonte de dados;
- O wrapper responde descrevendo que porção do fragmento pode ser aceita, uma estimativa do número de registros produzidos, do tempo de execução, etc.;
- O otimizador de consultas do servidor incorpora as respostas em um plano global, introduzindo operações adicionais para unir os vários fragmentos de consulta;

# Passo da Consulta – Estudo de Caso (cont.)

- As informações de custo e cardinalidade são utilizadas para estimar o custo total do plano. O plano de menor custo total é selecionado entre os candidatos.
- O candidato selecionado pode ser armazenado em um catálogo de consultas para ser usado outras vezes;
- Uma vez o plano selecionado, o banco de dados federado dirige a execução invocando os wrappers para executar o fragmento associado a ele;
- Os *wrappers* executam as operações necessárias sobre as fontes de dados (ex. traduzir a consulta na linguagem nativa);
- Os resultados são retornados para o servidor federado, que combina-os, executa algum processamento adicional e retorna o resultado final para a aplicação.



Arquitetura de um sistema federado da IBM

# Projeto de Banco de Dados

- 1) **Análise de requisitos**: Levantamento nas necessidades (dados e aplicações);
- 2) **Projeto conceitual**: Desenvolvimento de uma descrição de alto nível dos dados armazenados (diagrama ER);
- 3) **Projeto lógico**: Transformação do modelo ER em um esquema de base de dados relacional;
- 4) **Refinamento do esquema**: Análise do esquema relacional para identificar potenciais problemas (normalização);
- 5) **Projeto físico**: Refinamento do projeto para assegurar os critérios de performance desejados;
- 6) **Projeto de Segurança**: Definição dos usuários e permissões .

# Normalização

- A normalização serve como alternativa para o refinamento do esquema relacional, visando minimizar a redundância dos dados armazenados no banco de dados;
- Existem várias formas normais: 1ªFN, 2ªFN, 3ªFN e etc.;
- Cada relação na 3ªFN está também na 2ªFN, cada relação na 2ªFN está também na 1ªFN.

# Normalização (cont.)

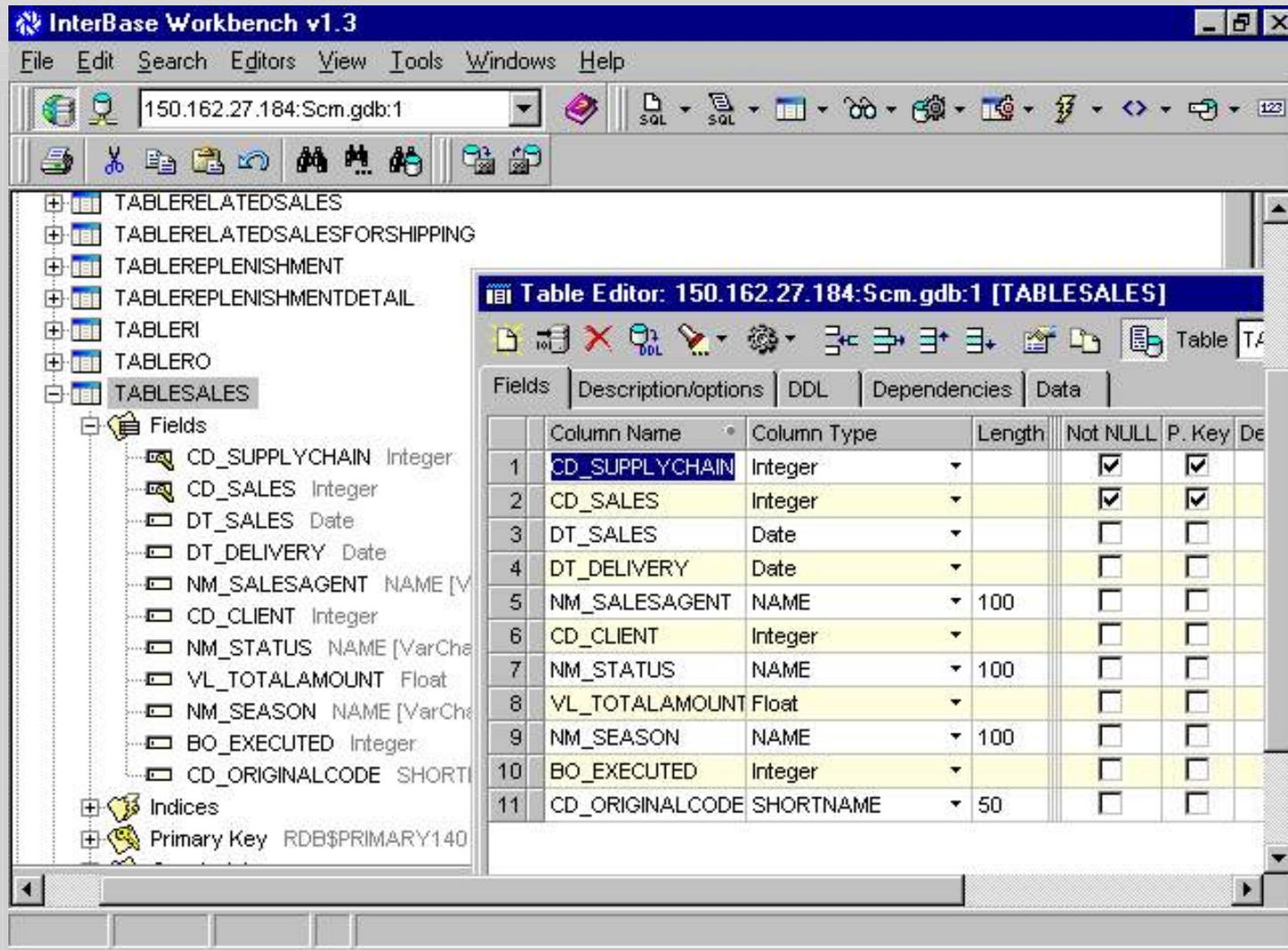
- **1ªFN:** Uma tabela está na 1ªFN se não possuir campos multivalorados;
- **2ªFN:** Uma tabela está na 2ªFN se estiver na 1ªFN e todos os seus campos dependam incondicionalmente da chave primária. Ou seja, não deve haver dependências parciais;
- **3ªFN:** Uma tabela está na 3ªFN se estiver na 2ªFN e não existir replicação de dados devido à existência de dependências transitivas.

# Escolha do BD

- O SGBD para implementação de um banco de dados deve:
  - Conter todos os requisitos exigidos pelas aplicações;
  - Ser compatível com a especificação SQL ANSI;
  - Ter controle de usuários e acesso concorrente;
  - Ter interfaces para criação de tabelas, views, procedures, triggers e sequências;
  - Ter interfaces para especificação de chaves primárias, chaves estrangeiras, constraints e domínios;
  - Variada biblioteca de *Drivers* para o acesso ao DB em diferentes linguagens e arquiteturas;
  - Custo reduzido (ou freeware);

# Banco de dados XML

- Existem dois tipos:
  - **Banco de dados integrados com XML:** São bancos de dados com extensões para transferir os seus dados para documentos XML e vice-versa. Ex: DB2, SQL Server 2000, Oracle 8i e 9i;
  - **Banco de dados XML Nativos:** São bancos de dados que armazenam XML na forma nativa. Ex: Tamino, XpSQL;
- A escolha de um ou outro tipo dependerá dos requisitos dos documentos XML utilizados.
  - Centrados em dados;
  - Centrados no documento.



Exemplo de ambiente de banco de dados

# Exercício

- Construa o modelo conceitual (ER) e o modelo lógico (normalizado) do banco de dados que armazene as seguintes informações:
  - Cadastro de Clientes;
  - Cadastro de Armazéns;
  - Cadastro de Produtos;
  - Lista de Pedidos.

