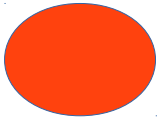
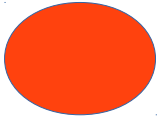
Modelos de dados

Aula 03
Prof. André Britto



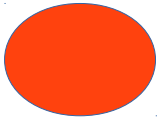
Abstração dos dados

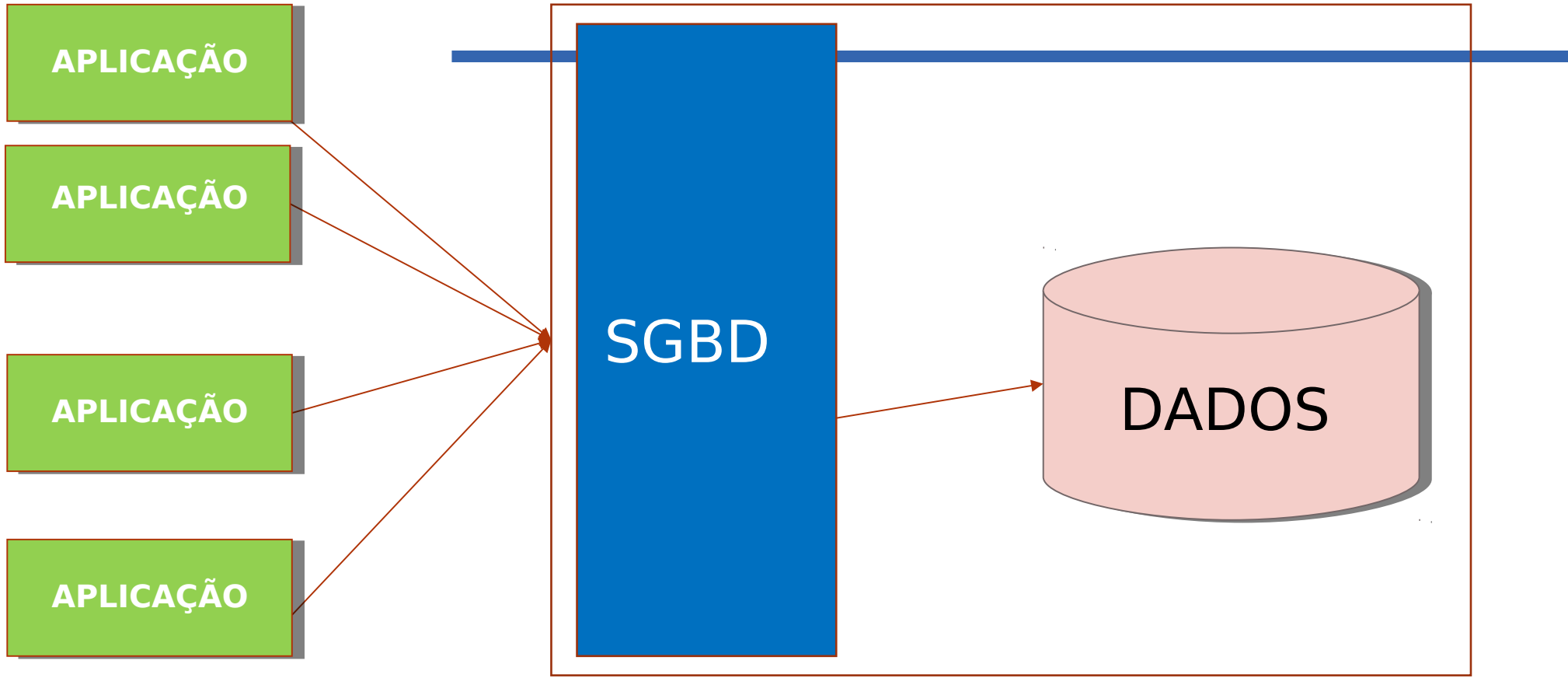
- Um dos principais propósitos de um sistema de banco de dados é prover ao usuário uma visão abstrata dos dados.
- O sistema esconde certos detalhes de como os dados são armazenados e mantidos.



Abstração dos dados

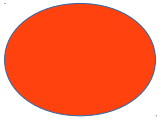
- Para que o sistema possa ser utilizado, ele deve recuperar dados de forma eficiente.
- Estruturas de dados complexas para representar dados em banco de dados.
- Os usuários não têm conhecimento de computação.
- Esconder a complexidade dos usuários através de vários níveis de abstração.




APLICAÇÃO
APLICAÇÃO
APLICAÇÃO
APLICAÇÃO
SGBD
DADOS

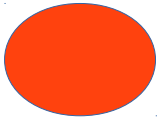
Modelo de dados

- Coleção de ferramentas conceituais para descrever a estrutura de um banco de dados.
- Provê os meios necessários para alcançar a abstração dos dados.
- Um modelo de dados provê um meio para descrever o projeto de um banco de dados nos níveis físico, conceitual e externo.

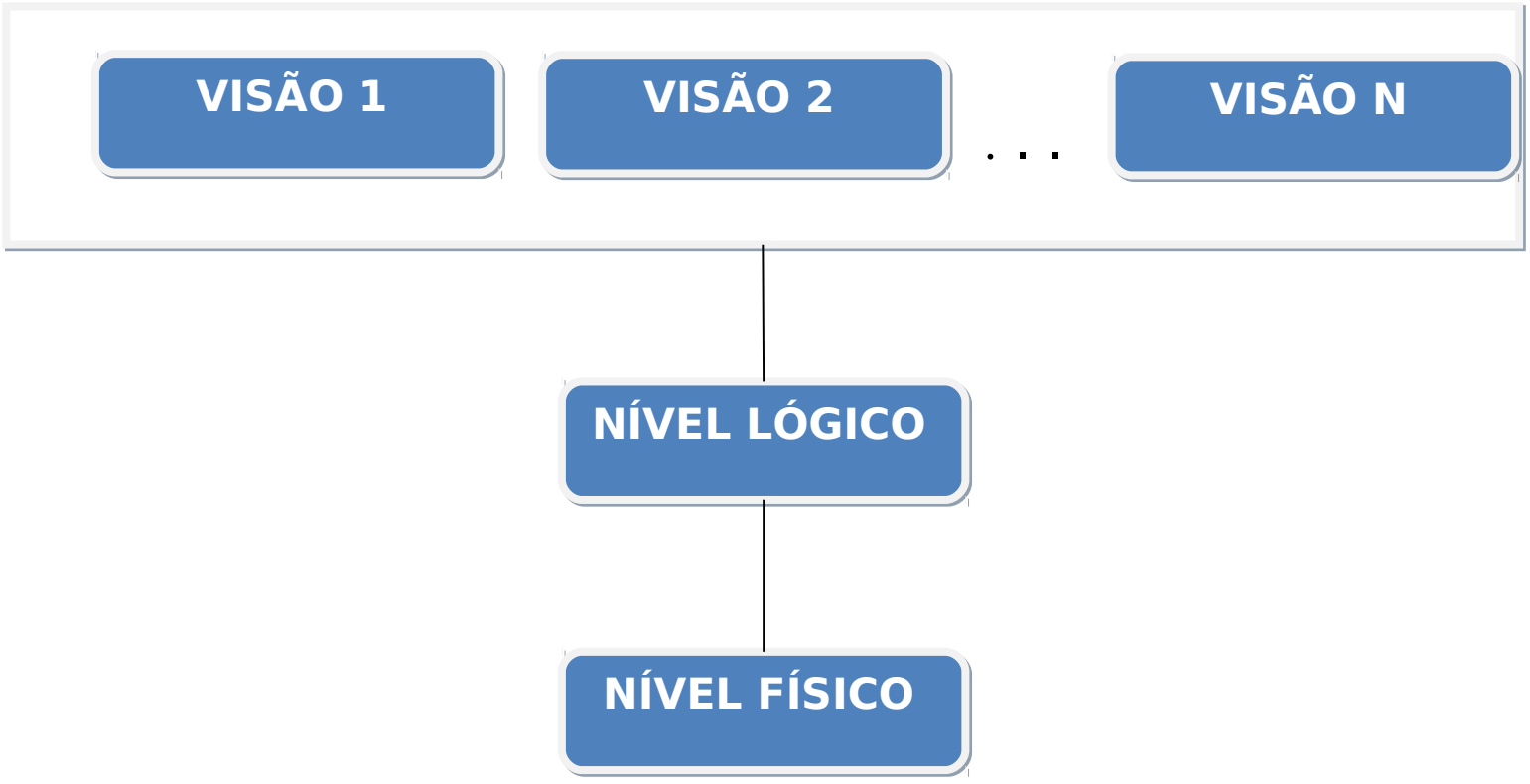


Modelo de dados

- Uma característica fundamental nesse contexto é a abstração dos dados.
  - Corresponde à supressão de detalhes da organização e armazenamento dos dados.
  - Busca enfatizar aspectos essenciais para um melhor entendimento dos dados.



Três níveis de abstração



Três níveis de abstração

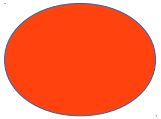
- Nível físico
  - É o nível de abstração mais baixo.
  - Descreve como os dados são realmente armazenados e caminhos de acesso para o banco de dados.

Três níveis de abstração

- Nível lógico
  - É uma camada acima do modelo físico e descreve quais dados estão armazenados.
  - Esconde os detalhes do armazenamento dos dados e descreve as entidades, os tipos de dados, relacionamentos, operações dos usuários e restrições.
  - Nível conceitual

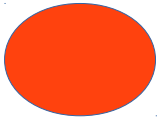
Três níveis de abstração

- Nível externo (Visões)
  - O nível mais alto de abstração, descreve somente partes do banco de dados que sejam de interesse de um determinado grupo de usuários.
  - Permite que o acesso aos dados seja customizável.



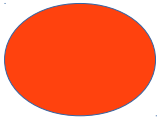
Independência de dados

- Independência lógica
  - Capacidade de se alterar o esquema do nível conceitual sem que haja alteração no nível externo.
  - Mapeamentos das visões para o esquema conceitual, assim, caso haja a alteração nesse nível só é necessária a redefinição dos mapeamentos.



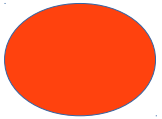
Independência de dados

- Independência física
  - Capacidade de se alterar o esquema físico sem mexer no esquema conceitual.
  - Mudanças no nível físico ocorrem quando é necessário reorganizar os dados para se obter uma melhora no desempenho do acesso e manipulação.



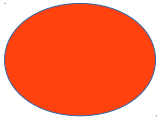
Modelo de dados

- Modelos de dados mais utilizados:
  - Modelo relacional e objeto-relacional.
  - Modelo baseado em objetos.
  - NoSQL.
  - Modelos legados
    - Rede e hierárquico.



Modelo relacional

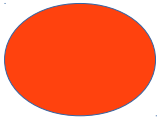
- É um modelo formal baseado no conceito das relações matemáticas
  - Teoria dos conjuntos.
- Esquema lógico (conceitual).
- Representa o banco de dados como uma coleção de relações.
- É definido por um conjunto de relações
  - Tabela de valores.



Modelo relacional

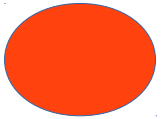
- Exemplos:
  - Estudantes de uma universidade

Nº Matrícula	Nome	Endereço	Telefone
100	André	Rua A	1111-1111
200	Beatriz	Rua B	2222-2222
300	Pedro	Rua C	3333-3333



Esquemas e instâncias

- Um banco de dados muda com o tempo à medida que novas informações são inseridas ou removidas.
- Instância do banco de dados.
  - A coleção dos dados armazenados num banco de dados num momento particular
- Esquema do banco de dados
  - Projeto geral do banco de dados que descreve de forma conceitual os dados armazenados.
  - Não é comumente alterado.



Banco de dados que suportam objetos

- Banco de dados relacionais armazenam uma coleção fixa de tipo de dados.
  - Inteiros, datas, string.
- Porém, hoje em dia trabalhamos cade vez mais com dados complexos.
- Os conceitos de orientação a objetos vem influenciando banco de dados a suportarem dados complexos.

Banco de dados que suportam objetos

- Bancos de dados que suportam objetos se desenvolveram em dois caminhos:
  - Banco de dados orientados a objetos (BDOO).
  - Banco de dados objeto-relacional (BDOR).

Sistemas de banco de dados objeto-relacional

- Tentativa de estender SGBD relacionais com funcionalidades para suportar uma classe mais larga de aplicações.
- Proveem uma ponte entre os paradigmas relacional e orientado a objetos.

Sistemas de banco de dados objeto-relacional

- Fornecem suporte para consultas complexas sobre dados complexos.
- Permite especificar e utilizar tipos abstratos de dados.
- Implementa conceitos como referência a objetos, herança e uso de construtores.

Sistemas de banco de dados objeto-relacional

- O resultado de uma consulta ainda consiste de tabelas .
- Um SGBD Objeto-Relacional ainda é relacional
  - Dados armazenados em tabelas formadas por linhas e colunas.
- A linguagem de consultas é uma extensão da linguagem SQL.
  - Padronizada na SQL 1999.

Sistemas de banco de dados objeto-relacional

- Criação de tipos
- Criação de métodos
- Herança entre tabelas
- Definição de um tipo array e multiset.

Sistemas de banco de dados orientados a objetos

- São uma alternativa aos bancos de dados relacionais.
- Destinados a aplicações onde objetos complexos tem um papel central.
- Influenciado pelas linguagens de programação orientadas a objeto.

Sistemas de banco de dados orientados a objetos

- Utiliza um modelo de dados diferente
  - Modelo de dados ODMG.
  - *Object Database Management Group* (Não existe mais).
- Linguagens de programação persistentes
  - Java Database Objects (JDO).

NoSQL

- *Not Only SQL*
- Surgiu de limitações de bancos de dados relacionais.
- Novos tipos de dados.
- Problemas de escalabilidade e desempenho.

NoSQL

- Categorias
  - Chave-valor.
  - Tabular.
  - Documentos.
  - Grafo.

NoSQL

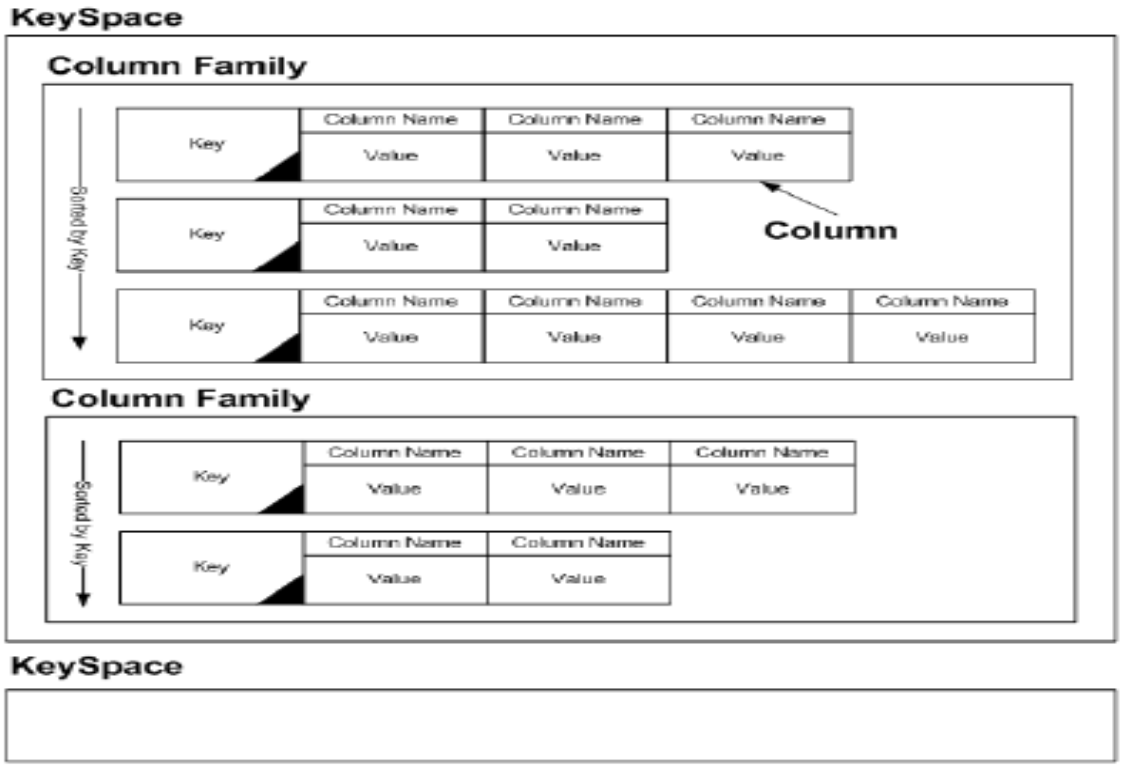
- Chave-valor
  - Similar a uma tabela hash.
  - Os dados tem uma chave e um valor associado.
  - Geralmente os valores são definidos como vetore s de bytes.
  - Acesso definido por métodos: get e put.

NoSQL

- Colunas/Tabular
  - Representa dados em tabelas, porém de forma bastante flexível.
  - Dividido em seções, que podem receber diferentes tabelas.
  - Dados são identificados por linhas, colunas e timestamps.

NoSQL

- Colunas/Tabular

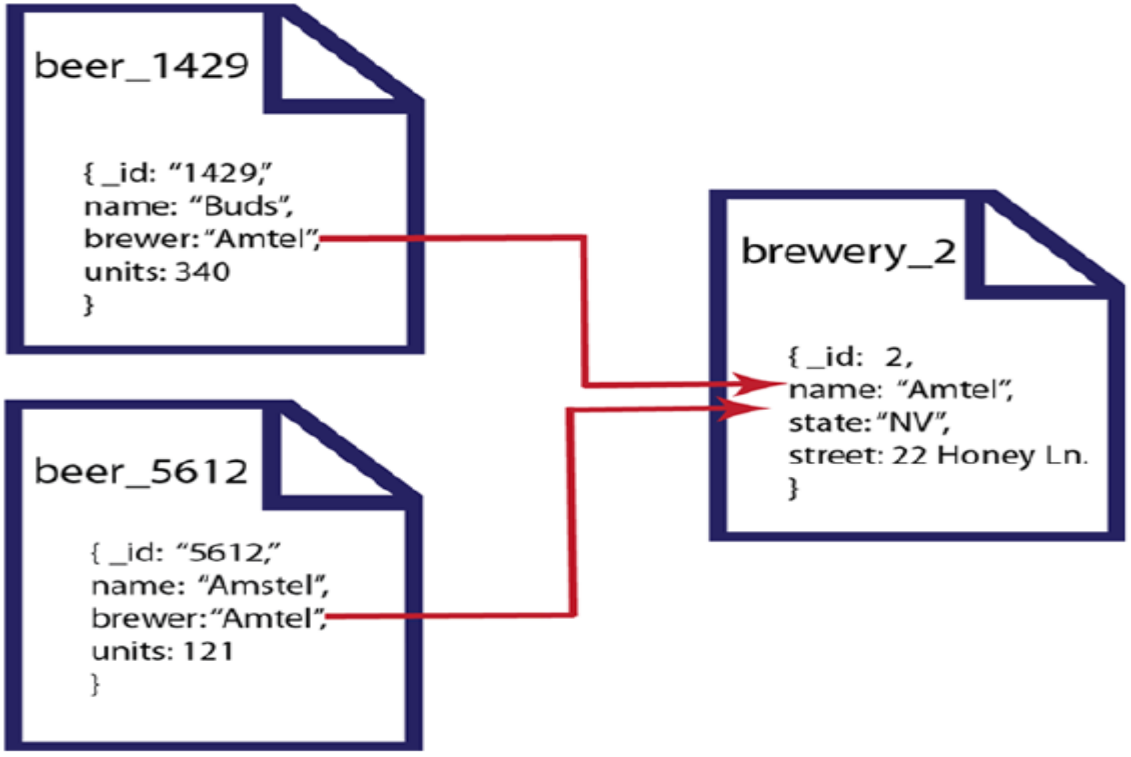


NoSQL

- Documentos
  - Uma forma mais desenvolvida de chave-valor.
  - O valor armazenado é estruturado em um documento.
  - Bancos de dados semi-estruturados.
  - Documentos
    - XML, JSON, dados binários, imagens, arquivos .pdf, arquivos de texto, etc.

NoSQL

- Documentos

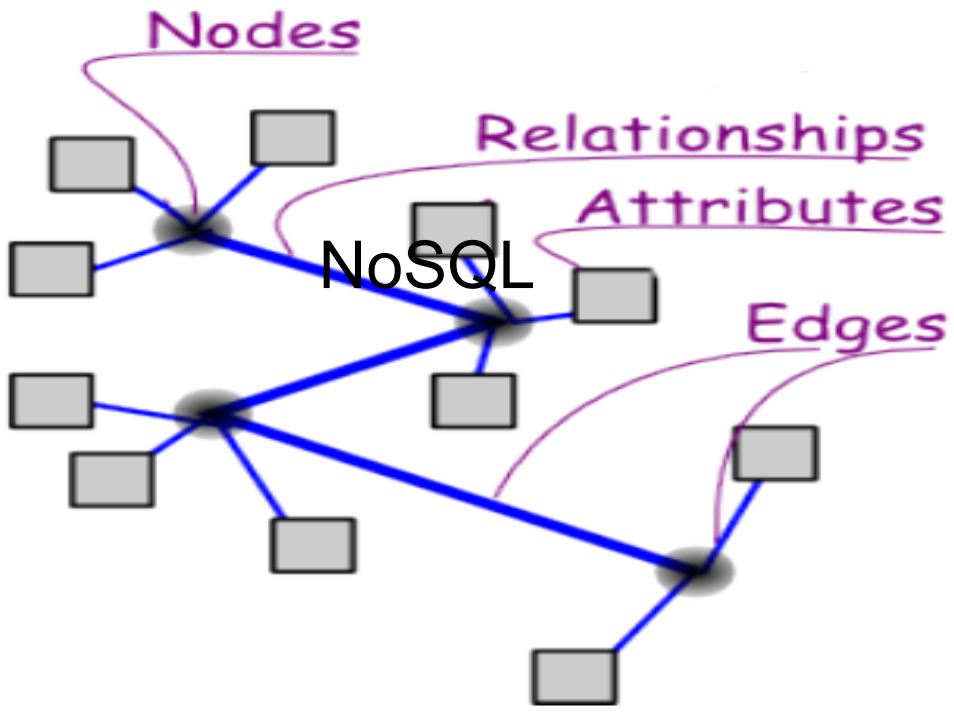


NoSQL

- Grafos
  - Utiliza conceito de grafos para representar os dados armazenados.
  - Modelo mais abstrato, porém, com maior liberdade para a representação.
  - Grafos rotulados e direcionados.

NoSQL

- Grafos

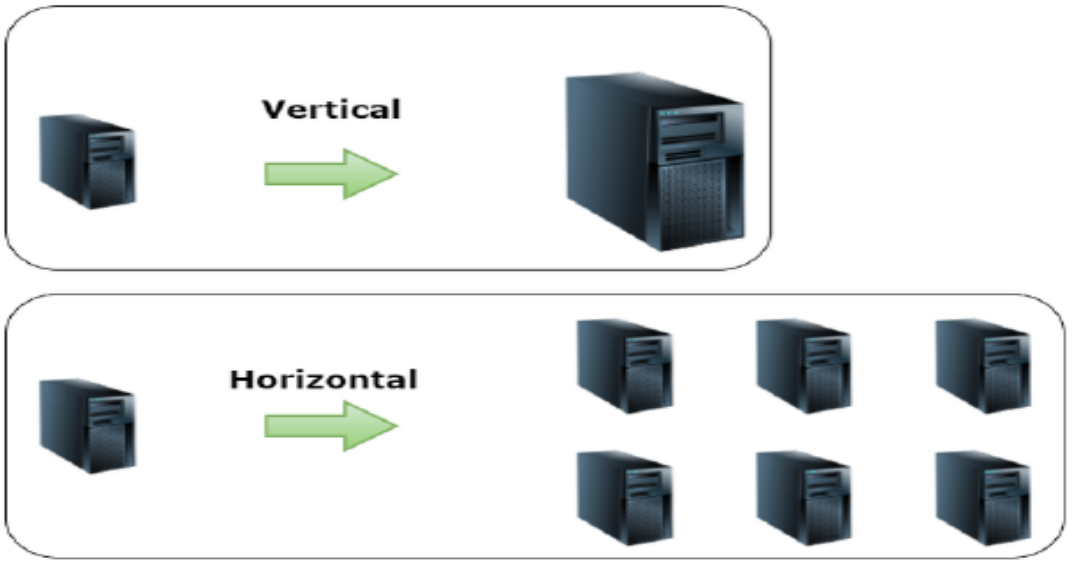


NoSQL

- Características
  - Não relacionais → contraponto à bancos relacionais.
  - Distribuído
    - Bancos de distribuídos possui componentes espalhados em diferentes locais físicos.
    - Bancos NoSQL tem como característica serem distribuídos.

NoSQL

- Características
  - Código aberto.
  - Escalável horizontalmente



NoSQL

- Características
  - Suporte à replicação.
  - API simples.
  - Esquema flexíveis.
    - Não é necessária a definição de um esquema fixo.

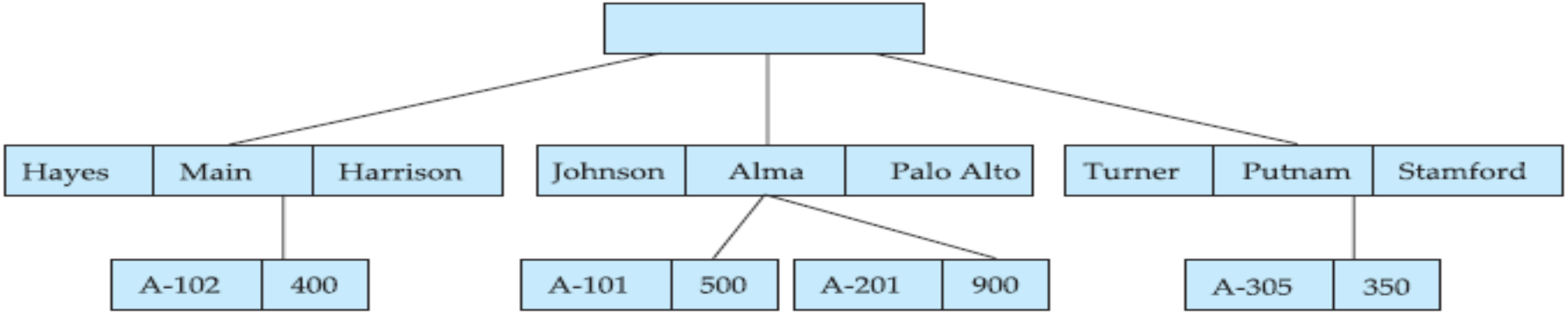
Modelos legado

- Modelo de rede
- Modelo hieráquico

Modelo hierárquico

- Consiste numa coleção de registros que estão conectados através de links.
- Um link também é a associação de somente dois registros.
- Os registros estão organizados através de uma árvore.

Modelo hierárquico



Modelo hierárquico

- Todos os registros estão organizados numa estrutura de árvore.
- A raiz é um nó falso.
- Um banco de dados hierárquico é uma coleção dessas raízes
  - Forma uma floresta.
- Árvore do banco de dados.

Diagrama de estrutura de árvores

- Representa o esquema de um banco de dados hierárquico.
- Esse diagrama consiste em:
  - Caixas – corresponde a tipo de registros.
  - Linhas – corresponde aos links.
- Especifica a a estrutura lógica do banco de dados.

Diagrama de estrutura de árvores

- Os relacionamentos formados devem ser do tipo um-para-muitos ou um-para-um entre pais e filhos.

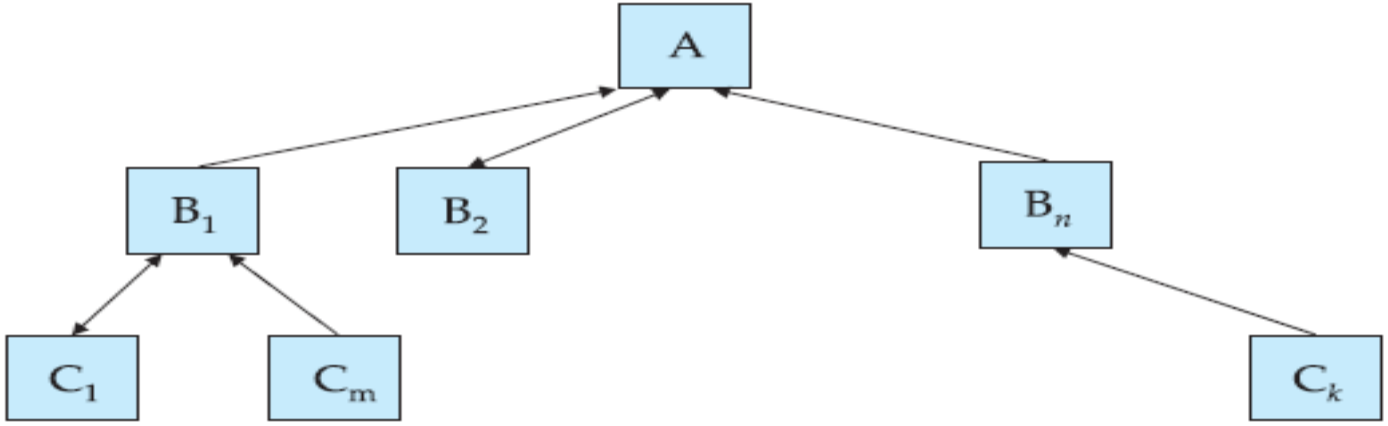


Diagrama de estrutura de árvores

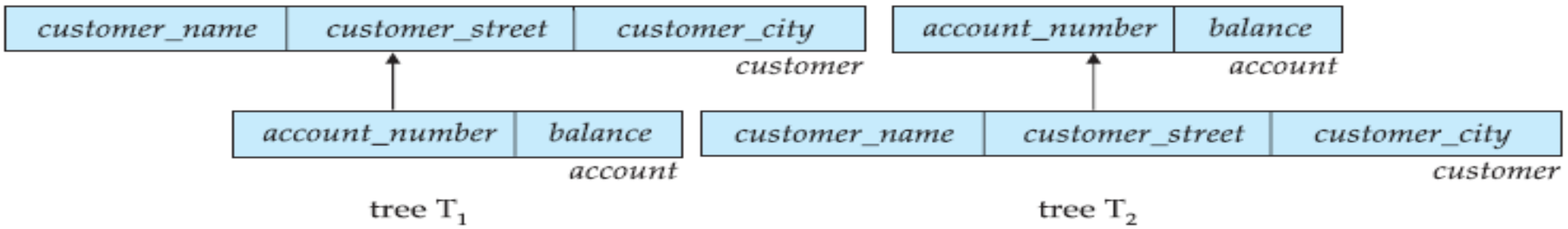
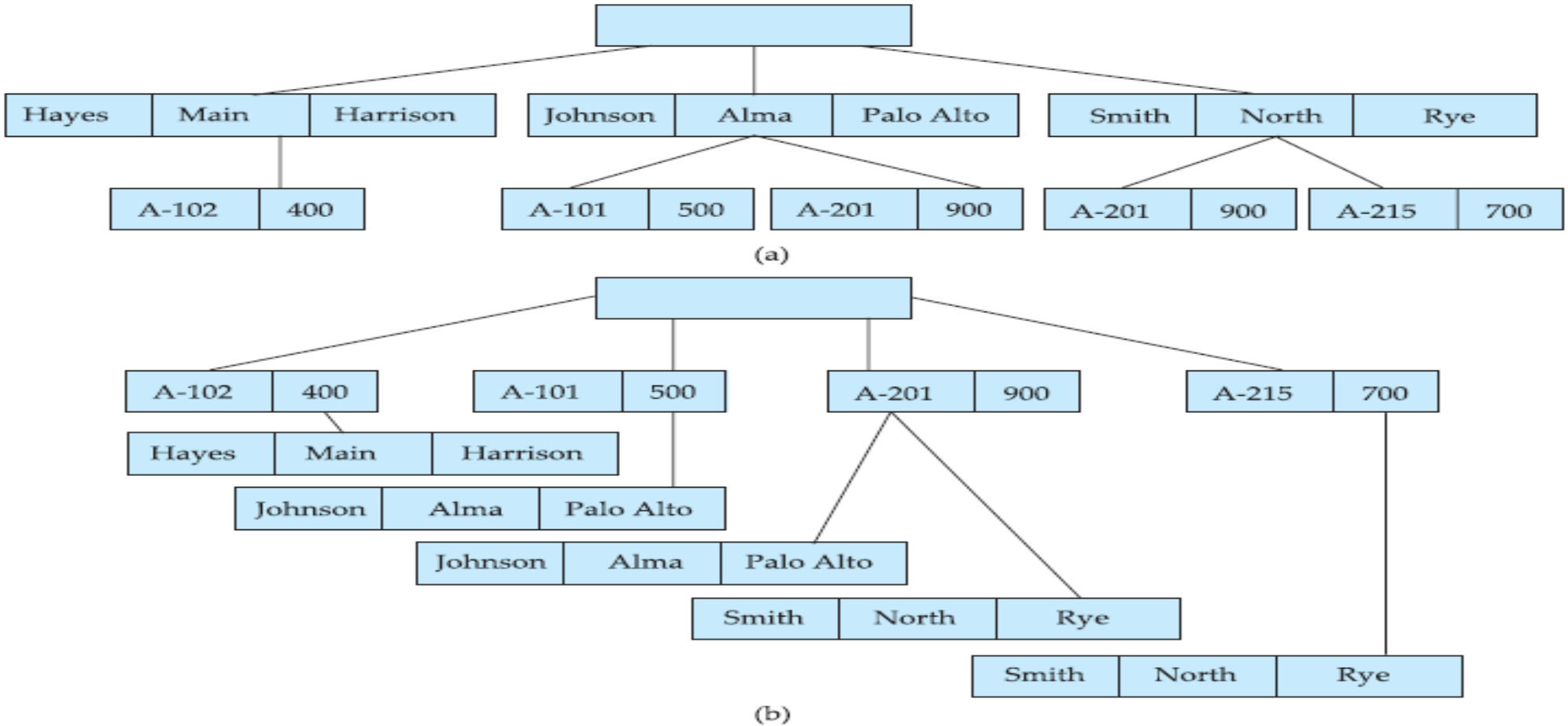


Diagrama de estrutura de árvores



Recuperação de informação

- Utiliza uma linguagem de consulta.
- DL/I
  - Linguagem de manipulação de dados do IMS.
- Utiliza comandos como get, get first, get next para percorrer a árvore.

Recuperação de informação

```
get first account
      where account.balance > 500;
while DB-status = 0 do
     begin
          print (account.account_number);
          get next account
               where account.balance > 500;
     end
```

Sistema de Banco de Dados IMS

- O modelo hierárquico é importante por causa do IMS.
  - Desenvolvido pela IBM.
- Um dos mais antigos e mais utilizados banco de dados.
- Primeiro banco de dados a tratar questões como concorrência, recuperação, integridade e processamento de consultas.

Leitura recomendada

- SILBERSCHATZ, A; Korth H.F.; Sudarshan S. **Sistemas de Banco de Dados**, Editora Campus, 6ª Edição.
  - Capítulo 22, Apêndices D e E.
- RAMAKRISHNAN R; GEHRKE J. **Sistemas de Gerenciamento de Banco de Dados**. Mcgraw-Hill Interamericana, 3ª Edição.
  - Capítulo 25.

Sistemas de banco de dados objeto-relacional

- Criação de tipos
  - É possível criar tipos complexos.

    Create type Nome as

    (primeiro nome varchar(20),

    sobrenome varchar(20)) final;

Sistemas de banco de dados objeto-relacional

- Criação de tipos

```
create table Estudante(
   nome Nome,
   data_nascimento date,
   ...
)
```

Sistemas de banco de dados objeto-relacional

- É possível também criar métodos
  - Na criação do tipo definimos o método através da palavra chave **method** [identificado do método].
  - O corpo do método é definido pelo comando **create instance method** [identificado do método].

Sistemas de banco de dados objeto-relacional

- Herança de tipo
  - Um BDOR disponibiliza a possibilidade de criar tipos com herança.
  - Assim, um tipo herda todos os atributos e métodos do tipo pai.
  - Utiliza a palavra chave **under**.

Sistemas de banco de dados objeto-relacional

- Herança de tabela
  - Também é possível criar tabelas que herdam campos de outras tabelas.
  - É possível especificar tabelas que herdam atributos base de uma tabela pai.
  - Útil para mapear as notações definidas no modelo entidade relacionamento estendido.

Sistemas de banco de dados objeto-relacional

- Outras características
  - Definição de um tipo array e multiset.

Sistemas de banco de dados orientados a objetos

- São uma alternativa aos bancos de dados relacionais.
- Destinados a aplicações onde objetos complexos tem um papel central.
- Influenciado pelas linguagens de programação orientadas a objeto.

Sistemas de banco de dados orientados a objetos

- Utiliza um modelo de dados diferente
  - Modelo de dados ODMG.
  - *Object Database Management Group* (Não existe mais).
- Linguagens de programação persistentes
  - Java Database Objects (JDO).

Sistemas de banco de dados orientados a objetos

- Possui uma linguagem de banco de dados.
  - Object data language (ODL).
  - Object query language (OQL).
- Existem outras linguagens para BDOO.
  - Diferentes fabricantes utilizam diferentes linguagens.

Sistemas de banco de dados orientados a objetos

- Consiste numa coleção de objetos.
- Todo objeto possui um Object ID.
  - Não utiliza o conceito de chaves do modelo relacional.

Sistemas de banco de dados orientados a objetos

- Um banco de dados contém um conjunto de objetos com propriedades semelhantes.
  - Classe.
- Essas propriedades são de três tipos:
  - Atributos.
  - Relacionamentos.
  - Métodos.

Sistemas de banco de dados orientados a objetos

- Atributos
  - Podem ser um tipo atômico.
  - Ou um tipo estruturado.
  - A ODL suporta *sets, bag, list, arrays* e *structs*.

Sistemas de banco de dados orientados a objetos

- Relacionamentos
  - É uma referência para um objeto;
  - Ou um coleção de referências.
  - Captura como um objeto está relacionado com um ou mais objetos da mesma classe ou de classes diferentes.

Sistemas de banco de dados orientados a objetos

- Métodos
  - Funções que são aplicadas a objeto de uma classe.
  - Não existe nenhuma função análoga em banco de dados relacionais.

Sistemas de banco de dados orientados a objetos

```
interface Movie
        (extent Movies key movieName)
        { attribute date start;
        attribute date end;
        attribute string moviename;
        relationship Set⟨Theater⟩ shownAt inverse Theater::nowShowing;
        }
```

Sistemas de banco de dados orientados a objetos

```
interface Theater
        (extent Theaters key theaterName)
        { attribute string theaterName;
        attribute string address;
        attribute integer ticketPrice;
        relationship Set⟨Movie⟩ nowShowing inverse Movie::shownAt;
        float numshowing() raises(errorCountingMovies);
        }
```

Sistemas de banco de dados orientados a objetos

- Object Query Language
  - Linguagem de consulta de banco de dados orientados a objeto.
  - Semelhante à SQL.

Sistemas de banco de dados orientados a objetos

- Object Query Language

```
SELECT mname: M.movieName, tname: T.theaterName
FROM Movies M, M.shownAt T
WHERE T.numshowing() > 1
```

Banco de dados que suportam objetos

- BDOR X BDOO
  - Ambos suportam:
    - A criação de tipos abstratos de dados;
    - Identificação do objeto;
    - Tipos de referência;
    - Herança.
  - Ambos possuem linguagens de consulta:
    - Extensão da SQL e ODL/OQL.

Banco de dados que suportam objetos

- BDOR X BDOO
  - BDOR tentam adicionar funcionalidades OO em banco de dados relacionais.
  - BDOO desenvolveram linguagens de consultas baseadas em banco de dados relacionais.
  - Ambos SGBDs garantem funcionalidades como controle de concorrência e recuperação de falhas.

Banco de dados que suportam objetos

- BDOR X BDOO
  - A grande diferença e na maneira que eles são baseados.
  - BDOR adiciona novos tipos de dados em relação a banco de dados relacionais.
  - BDOO tenta adicionar funcionalidades de um SGBD numa linguagem de programação.

Banco de dados que suportam objetos

- BDOR X BDOO
  - BDOO visa a integração com linguagens OO.
  - Essa integração não é importante para BDOR.
  - BDOO é direcionada para aplicações onde o uso de objetos é foco.
    - Consiste em recuperar e trabalhar com objetos.
  - BDOR é otimizada para trabalhar com coleção de dados grandes.

Modelo de rede

- Consiste numa coleção de registros conectados através de links.
- Cada registro é uma coleção de atributos.
  - Cada atributo contém um valor.
- Um link é um relacionamento binário entre dois registros.

Modelo de rede

- Como exemplo, vamos considerar um banco de dados representando a relação cliente (customer) conta (account) num sistema bancário.
- Temos dois tipos de registros:
  - Customer.
  - Account.
- Vamos usar uma notação semelhante à linguagem Pascal.

Modelo de rede

```
type customer = record
    customer_name = string;
    customer_street = string;
    customer_city = string;
    end
```

```
type account = record
    account_number = string;
    balance = string;
    end
```

Modelo de rede

- A figura representa que Hayes tem a conta A-102 e assim sucessivamente.

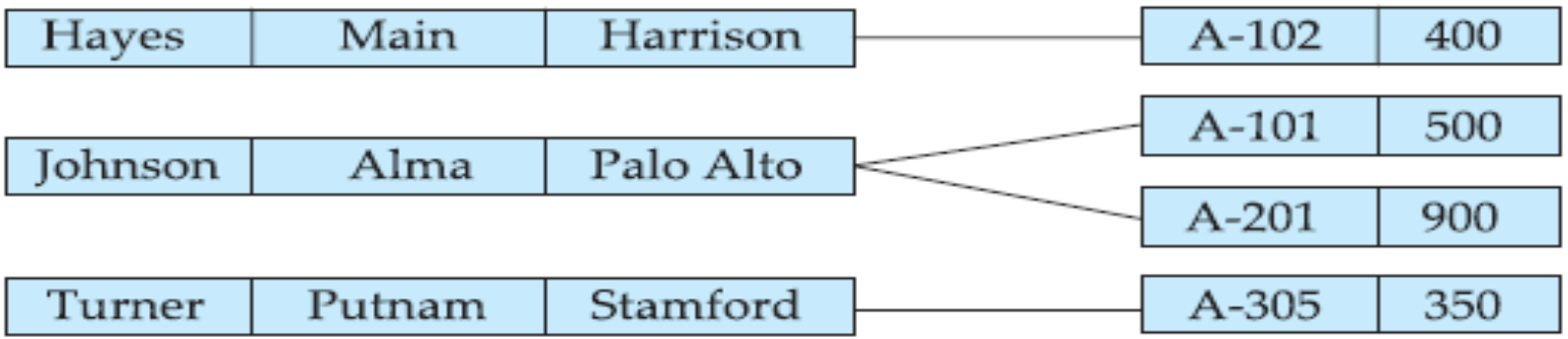
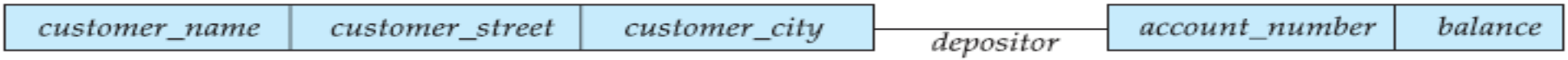


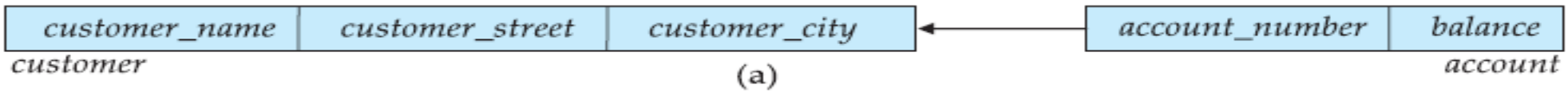
Diagrama de estrutura de dados

- Um diagrama de estrutura de dados representa o esquema de um banco de dados de redes.
- Esse diagrama consiste em:
  - Caixas – corresponde a tipo de registros.
  - Linhas – corresponde aos links.

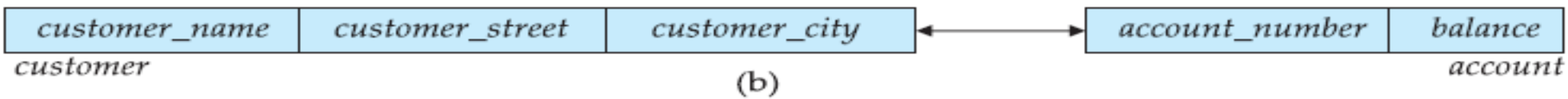
Relacionamentos binários



- Relacionamento muitos para muitos.

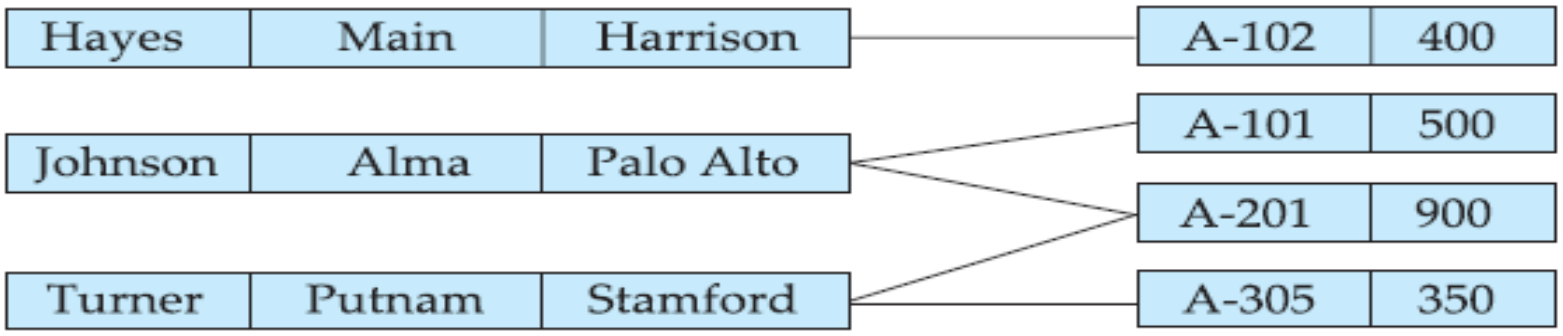


- Relacionamento um para muitos.



- Relacionamento um para um.

Relacionamentos binários



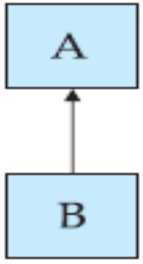
- Exemplo de um relacionamento muitos para muitos.

Modelo DBTG Codasyl

- Primeira especificação de um banco de dados.
  - Criado pelo Database Task Group nos anos 70
- Restrição de links
  - Permite somente relacionamentos muitos-para-um.
  - Os demais relacionamentos não são permitidos para facilitar a implementação do modelo.

Conjuntos DBTG

- Uma estrutura de dados consistindo em dois registros ligados.



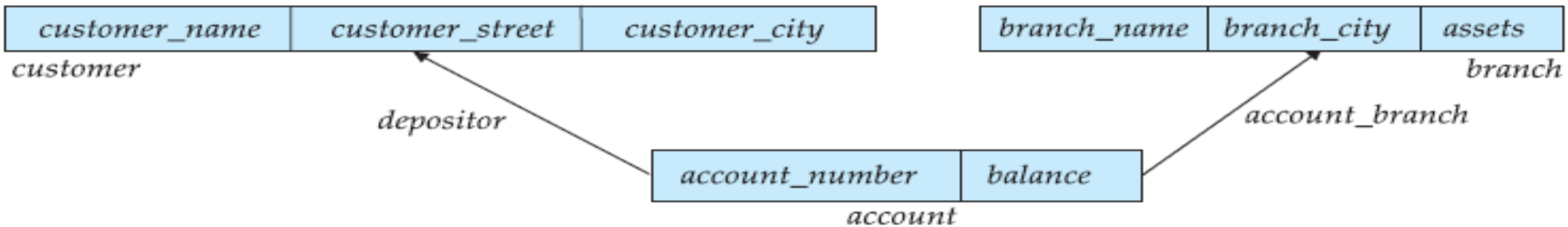
- O nome do conjunto é o mesmo do relacionamento entres os registros.

Conjuntos DBTG

- Como tipos de relacionamentos muitos-para-muitos não são permitidos, cada conjunto possui um proprietário e um membro.
- Nenhum membro pode participar de mais de um mesmo conjunto DBTG.
- Porém, um membro pode fazer parte de outros conjuntos DBTG.

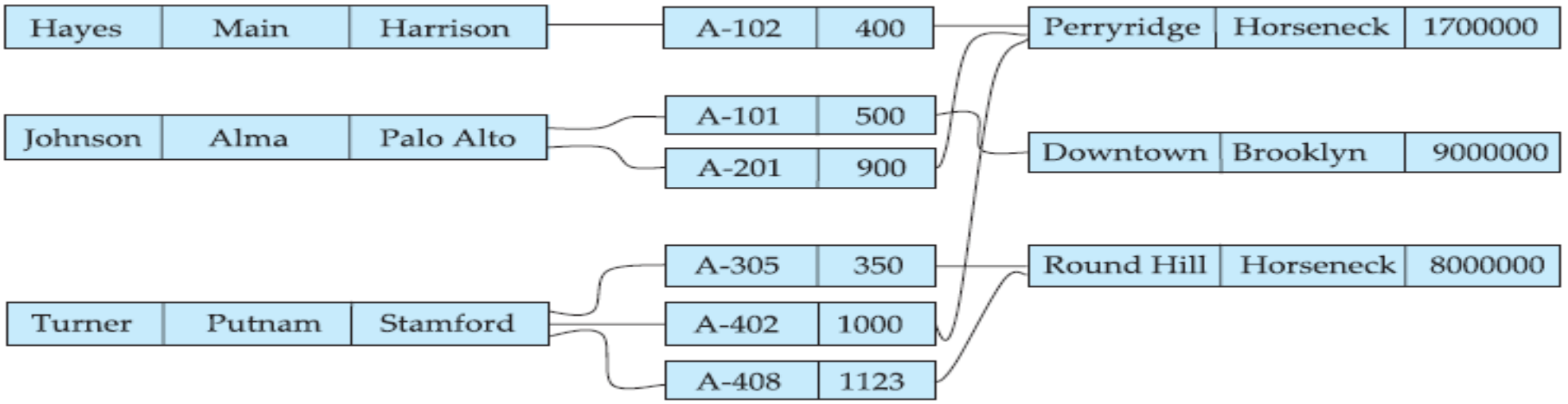
Conjuntos DBTG

- Dado diagrama de estrutura de dados:



Conjuntos DBTG

- Dado diagrama de estrutura de dados:



Conjuntos DBTG

- O banco de dados possui os seguintes conjuntos:

Set name is **depositor**
owner is **customer**
member is **account**

Set name is **account_branch**
owner is **branch**
member is **account**

Recuperação de dados

- A linguagem de manipulação consiste em comandos embutidos em linguagens de programação.
- O banco de dados mantém ponteiros para os registros acessados mais recentemente.
  - Um ponteiro para cada tipo de registro.
- Utiliza comandos como *find* e *get* para iterar sobre os registros armazenados.

Recuperação de dados

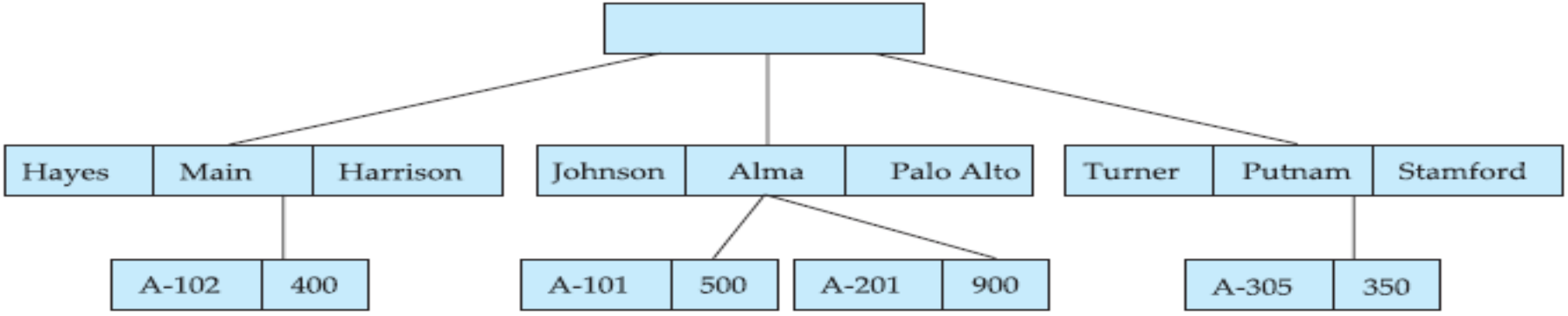
-

```
count := 0;
branch.branch_name := "Perryridge";
find any branch using branch_name;
find first account within account_branch;
while DB-status = 0 do
      begin
            get account;
            if account.balance > 10000 then count := count + 1;
            find next account within account_branch;
      end
print (count);
```

Modelo hierárquico

- Semelhante ao modelo de rede.
- Consiste numa coleção de registros que estão conectados através de links.
- Um registro é similar ao do modelo de rede.
- Um link também é a associação de somente dois registros.
- A diferença é que os registros estão organizados através de uma árvore.

Modelo hierárquico



Modelo hierárquico

- Todos os registros estão organizados numa estrutura de árvore.
- A raiz é um nó falso.
- Um banco de dados hierárquico é uma coleção dessas raízes
  - Forma uma floresta.
- Árvore do banco de dados.

Modelo hierárquico

- O conteúdo de um registro pode ser replicado em diferentes localizações.
  - Uma conta pode pertencer a mais de uma pessoa.
  - A informação deverá ser replicada.
- Essa replicação pode ocorrer numa mesma árvore ou em árvores diferentes.

Diagrama de estrutura de árvores

- Representa o esquema de um banco de dados hierárquico.
- Esse diagrama consiste em:
  - Caixas – corresponde a tipo de registros.
  - Linhas – corresponde aos links.
- Especifica a a estrutura lógica do banco de dados.

Diagrama de estrutura de árvores

- No modelo de redes, o diagrama de estrutura de dados representa um grafo arbitrário.
- No modelo hierárquico, o diagrama utilizado representa uma árvore com raiz.
- Em um árvore não podem haver ciclos.

Diagrama de estrutura de árvores

- Os relacionamentos formados devem ser do tipo um-para-muitos ou um-para-um entre pais e filhos.

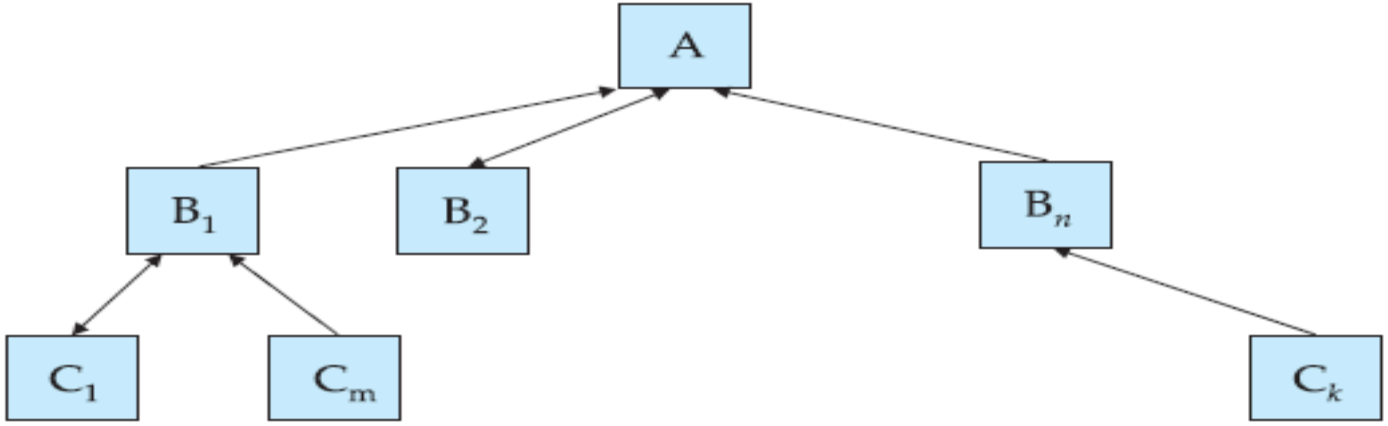
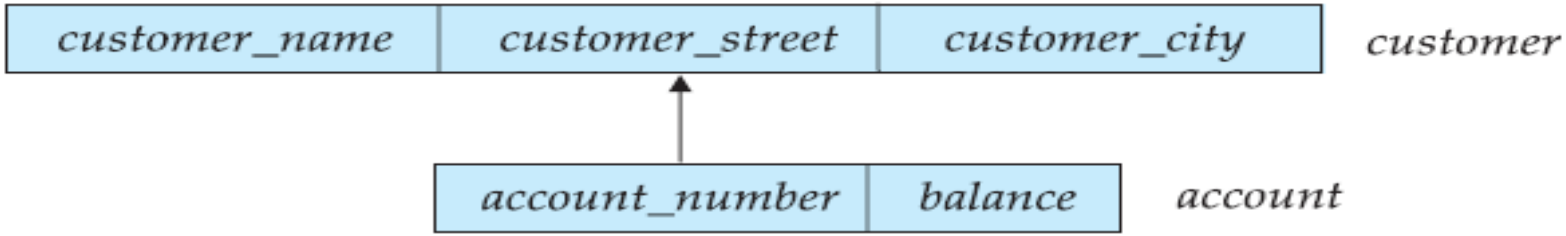
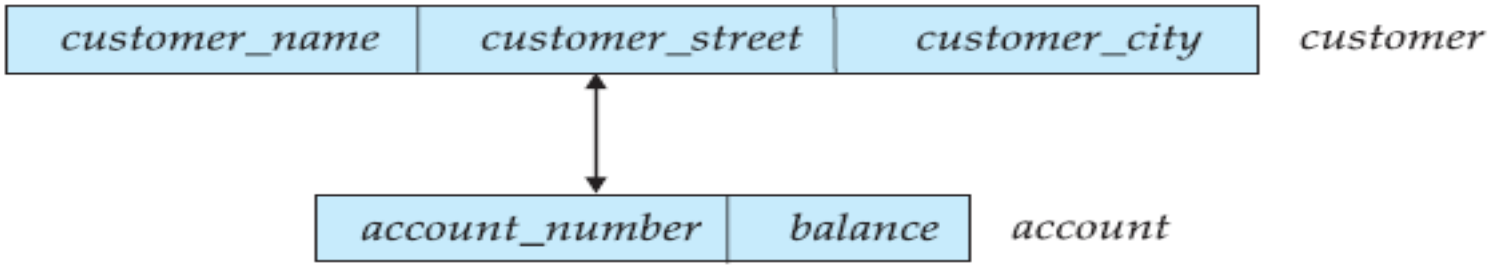


Diagrama de estrutura de árvores



- Relacionamento um para muitos.



- Relacionamento um para um.

Diagrama de estrutura de árvores

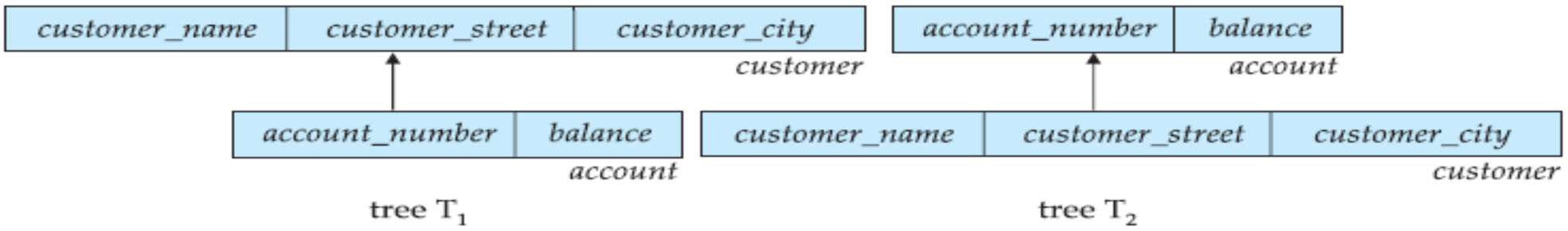
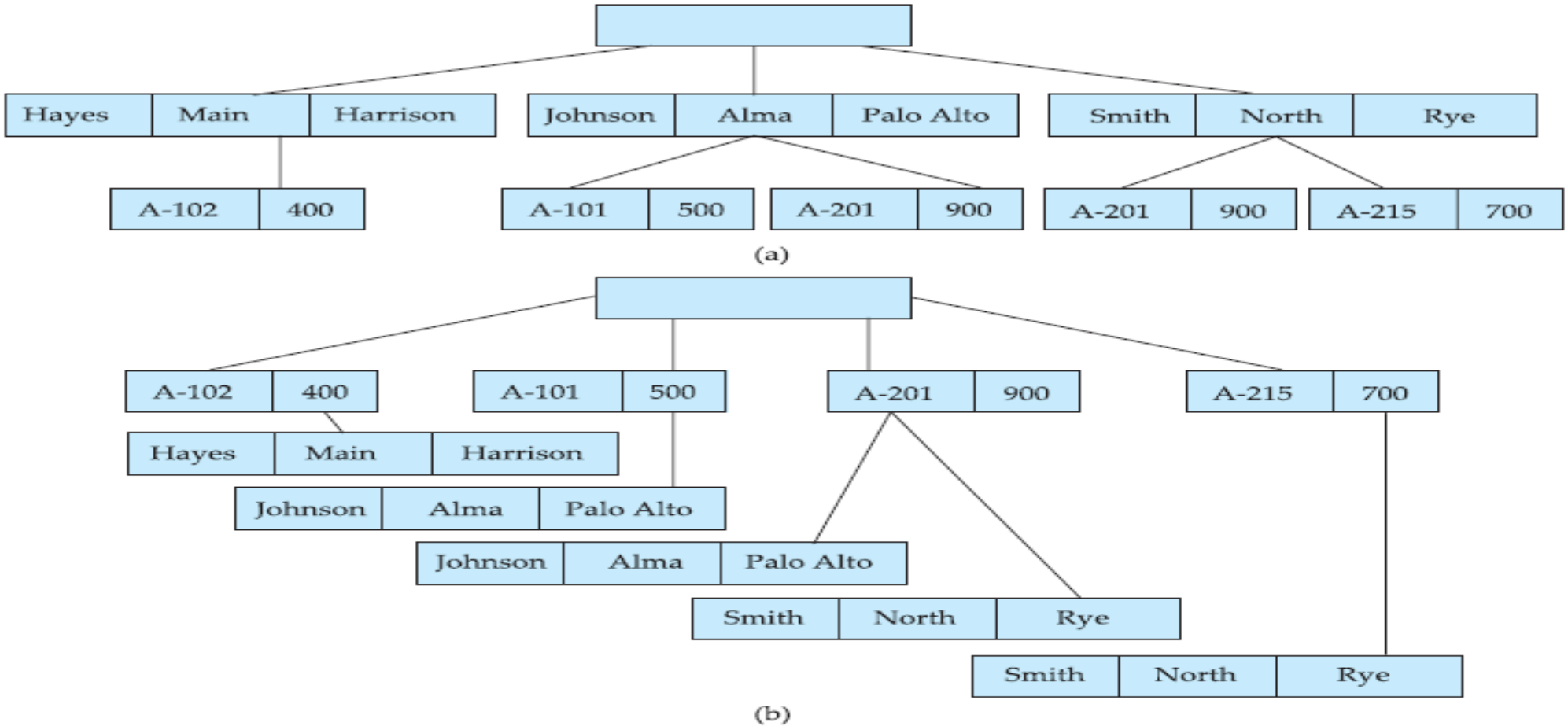


Diagrama de estrutura de árvores



Recuperação de informação

- Utiliza uma linguagem de consulta.
- DL/I
  - Linguagem de manipulação de dados do IMS.
- Utiliza comandos como get, get first, get next para percorrer a árvore.

Recuperação de informação

```
get first account
      where account.balance > 500;
while DB-status = 0 do
     begin
          print (account.account_number);
          get next account
               where account.balance > 500;
     end
```

Sistema de Banco de Dados IMS

- O modelo hierárquico é importante por causa do IMS.
  - Desenvolvido pela IBM.
- Um dos mais antigos e mais utilizados banco de dados.
- Primeiro banco de dados a tratar questões como concorrência, recuperação, integridade e processamento de consultas.

Modelo multi-dimensional

- Diferentemente do modelo relacional que guarda valores de atributos numa tabela de dados, o modelo multidimensional foca numa **coleção de medidas numéricas**.
- Essas medidas dependem de um conjunto de **dimensões**.

Modelo multi-dimensional

- Representa os dados através de um *array* multidimensional.
  - Cubo de dados.

Modelo multi-dimensional

- Exemplo: Vendas de uma loja.
  - Atributo medida → Número de vendas.
  - Dimensões → Produto, localização e tempo.
  - Dado um produto, uma localização e um tempo podemos ter associado um número de vendas.
  - Produto identificado por *pid*.
  - Tempo caracterizado por um *timeid*.
  - Localização caracterizada por um *locid*.

Modelo multi-dimensional

- *Array* multi-dimensional.

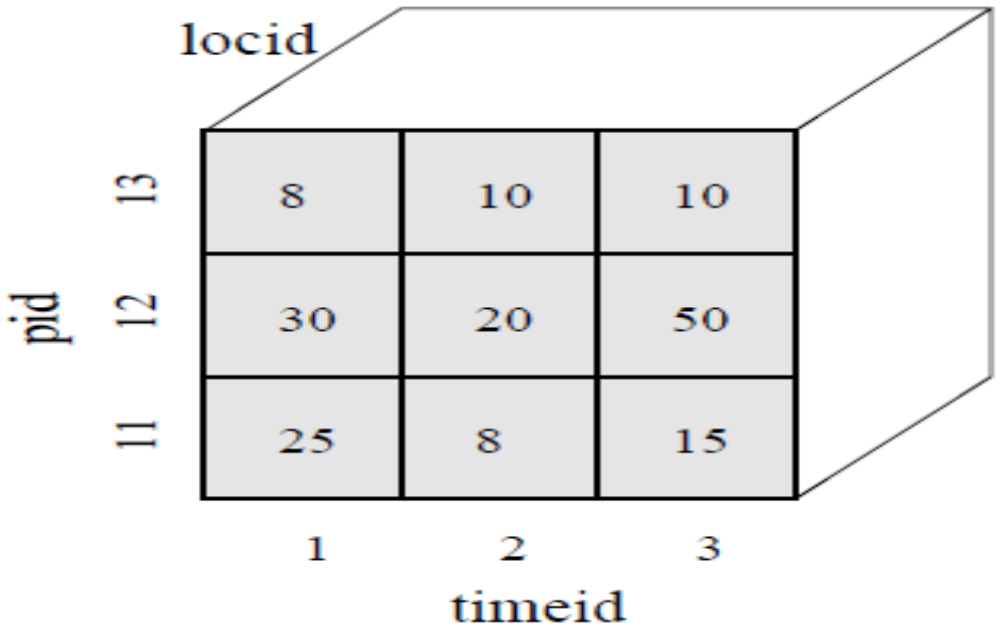


Figura 1. Array multi-dimensional. Cubo de dados.

Modelo multi-dimensional

- Essa visão de dados pode ser generalizada em mais de três dimensões.
  - Hipercubo de dados.
- Em aplicações OLAP os dados de fato podem estar organizados em *arrays* multi-dimensionais.
- Alguns banco de dados utilizam o modelo multi-dimensional.
  - MOLAP – Multidimensional OLAP systems.

Tabulação cruzada (Tabela pivô)

- Tabulação cruzada (Tabela pivô)
  - Tabela que relaciona valores de duas dimensões.
  - Valores das dimensões forma o cabeçalho das linhas e das coluna.
  - Valores das células é um valor do atributo medida.

Tabulação cruzada (Tabela pivô)

- Seja A o conjunto de todos os valores para uma dimensão.
  - ai é um valor do conjunto A.
- Seja B o conjunto de todos os valores para outra dimensão.
  - bj é um valor do conjunto B.
- (ai,bj) contém o valor do atributo medida para associação do valor ai com o valor bj.

Tabulação cruzada (Tabela pivô)

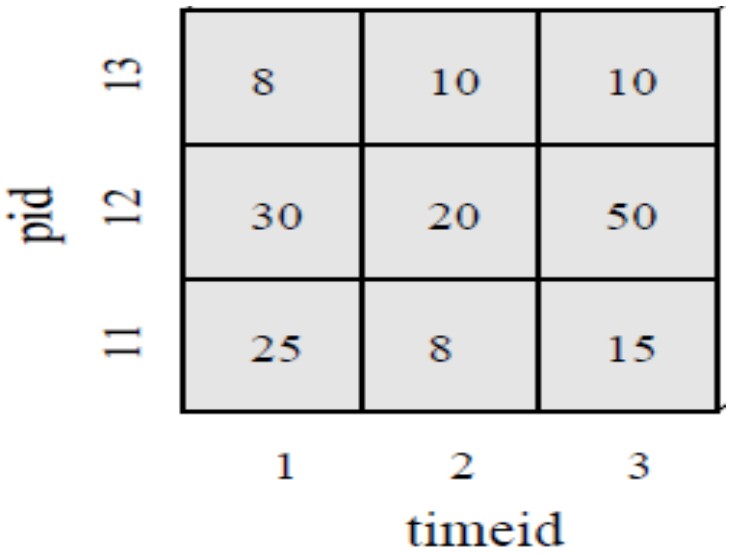


Figura 2. Tabela cruzada entre as dimensões produto e tempo
Ramakrishnam (2011).

Cubo de dados (hipercubos de dados)

- Cubo de dados (hipercubos de dados)
  - A generalização da tabela cruzada para *n* dimensões é chamada de cubo de dados (hipercubo de dados).
  - Um tabela cruzada pode ser vista como uma fatia do cubo de dados quando escolhemos uma dimensão.
  - No nosso exemplo, a tabela cruzada da Figura 2 pode ser obtida quando fixamos o valor de locid = 1.

Cubo de dados (hipercubos de dados)

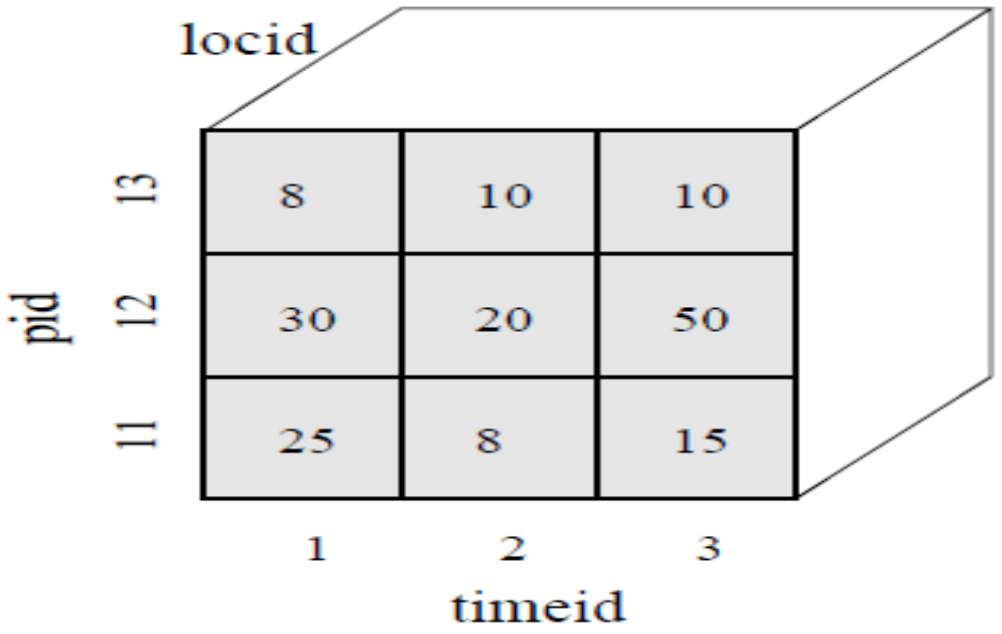


Figura 3. Cubo de dados.

Modelo multi-dimensional

- Modelo utilizado em sistemas de apoio à decisão ou sistemas BI.
- Modelo utilizado na criação de data warehouses.
- Tem foco na análise dos dados.
  - OLAP.

Modelo multi-dimensional

- Geralmente é representado no modelo relacional.
  - Tabela de fatos e tabela de dimensões.
  - ROLAP.

pid	timeid	locid	vendas
11	1	1	25
11	2	1	8
11	3	1	15
12	1	1	30
12	2	1	20
12	3	1	50
13	1	1	8
13	2	1	10
13	3	1	10
11	1	2	13
11	2	2	11
11	3	2	3
12	1	2	9
12	2	2	52
12	3	2	33
13	1	2	24

Tabela de dimensões

pid	Categoria	preço
11	1	20,00
12	1	30,00
13	2	100,00

Produto

timeid	Hora	dia	mês	ano
1	18:00	21	02	2012
2	19:00	21	02	2012
3	20:00	21	03	2012

Tempo

locid	bairro	cidade	estado	país
1	Atalaia	Aracaju	SE	Brasil
2	Centro	Curitiba	PR	Brasil
3	Jardins	Aracaju	SE	Brasil

Localização

Pontos importantes

- Modelos de dados descrevem de forma abstrata um esquema de banco de dados.
- Existem diversos modelos de dados.
- Os modelos de dados legados foram definidos juntamento com os primeiros SGBD.
  - Utilizados pode muito tempo.
  - Semelhantes ao modelo físico.

Pontos importantes

- Novos modelos de dados têm sido propostos.
- Estender o modelo relacional para tipos complexos.
- Mais adequados para linguagens de programação OO.

Pontos importantes

- DB4O
- Caché
  - Pós-relacional.
- MongoDB
- Oracle 11g
  - Relacional.
  - Objeto-relacional.
  - Orientado a objeto.