MAC0439
Laboratório de Bancos de Dados

Dados Semiestruturados

Profa. Kelly Rosa Braghetto
DCC-IME-USP

03 de agosto de 2016

Dados não estruturados

- Podem ser de qualquer tipo
- Não seguem necessariamente um formato ou sequência
- Não seguem regras
- Não são previsíveis
- **Exemplos:** texto, vídeo, som, imagens, ...

Dados estruturados

- São organizados em blocos semânticos (entidades)
- Entidades similares são mantidas de forma agrupada (relações ou classes)
- Entidades de um mesmo grupo possuem as mesmas descrições (atributos)
- As descrições para todas as entidades de um grupo (esquema) possuem o mesmo formato, o mesmo tamanho, estão todas presentes, e seguem a mesma ordem

Dados semiestruturados

- Os dados são organizados como entidades semânticas
- Entidades similares são mantidas de forma agrupada
- Entidades em um mesmo grupo podem não ter os mesmos atributos
- A ordem dos atributos não é necessariamente importante
- Nem todos os atributos são obrigatórios
- O tamanho e o tipo de um atributo pode variar dentro de um mesmo grupo

Dados para acesso eletrônico

- Dados mantidos em um **Sistema Gerenciador de Banco de Dados (SGBD) Relacional** são **dados estruturados** – possuem uma estrutura de representação (= esquema) rígida, previamente projetada
- Boa parte dos dados disponíveis para acesso eletrônico **não estão mantidos em BDs em SGBDs Relacionais**
- Principal exemplo: **dados da web**, que geralmente possuem uma organização bastante heterogênea
  - mistura textos sem nenhuma informação com conjuntos de registros bem formatados
  - grande volume
  - muitos relacionamentos

Dados heterogêneos

- A heterogeneidade dificulta as operações de consulta aos dados
- Não há um esquema uniforme, a partir do qual as consultas possam ser formuladas
- Isso implica na necessidade de se realizar buscas de "alto custo"
  - buscas exaustivas nos dados
  - buscas por palavras-chaves (por meio de técnicas de recuperação de informação)
- Dados como esses da web são dados **semiestruturados**

Modelos de dados semiestruturados

- São capazes de representar tanto dados bastante estruturados quanto dados sem estruturação alguma
- São capazes de representar dados irregulares; algumas ocorrências de dados podem possuir informações incompletas ou complementares com relação a outras
- **Modelo de dados relacional** – há distinção clara entre o tipo dos dados (= esquema do BD) e os dados propriamente ditos (= instâncias).
- **Modelo de dados semiestruturado** – separação não é tão clara; **os dados são autodescritivos** – a informação do esquema está misturada com valores dos dados

Dados semiestruturados e a Web

- O modelo de dados semiestruturado é um padrão para a representação e troca de dados na web
- Ele trouxe melhorias importantes para a publicação e reúso de dados eletrônicos, por prover uma **sintaxe simples para os dados** que é, ao mesmo, **facilmente processável por máquinas** e **legível aos usuários**
- Além disso, a flexibilidade da tipagem em dados semiestruturados se tornou essencial para a integração de dados, especialmente na integração de dados heterogêneos por meio de sistemas mediadores

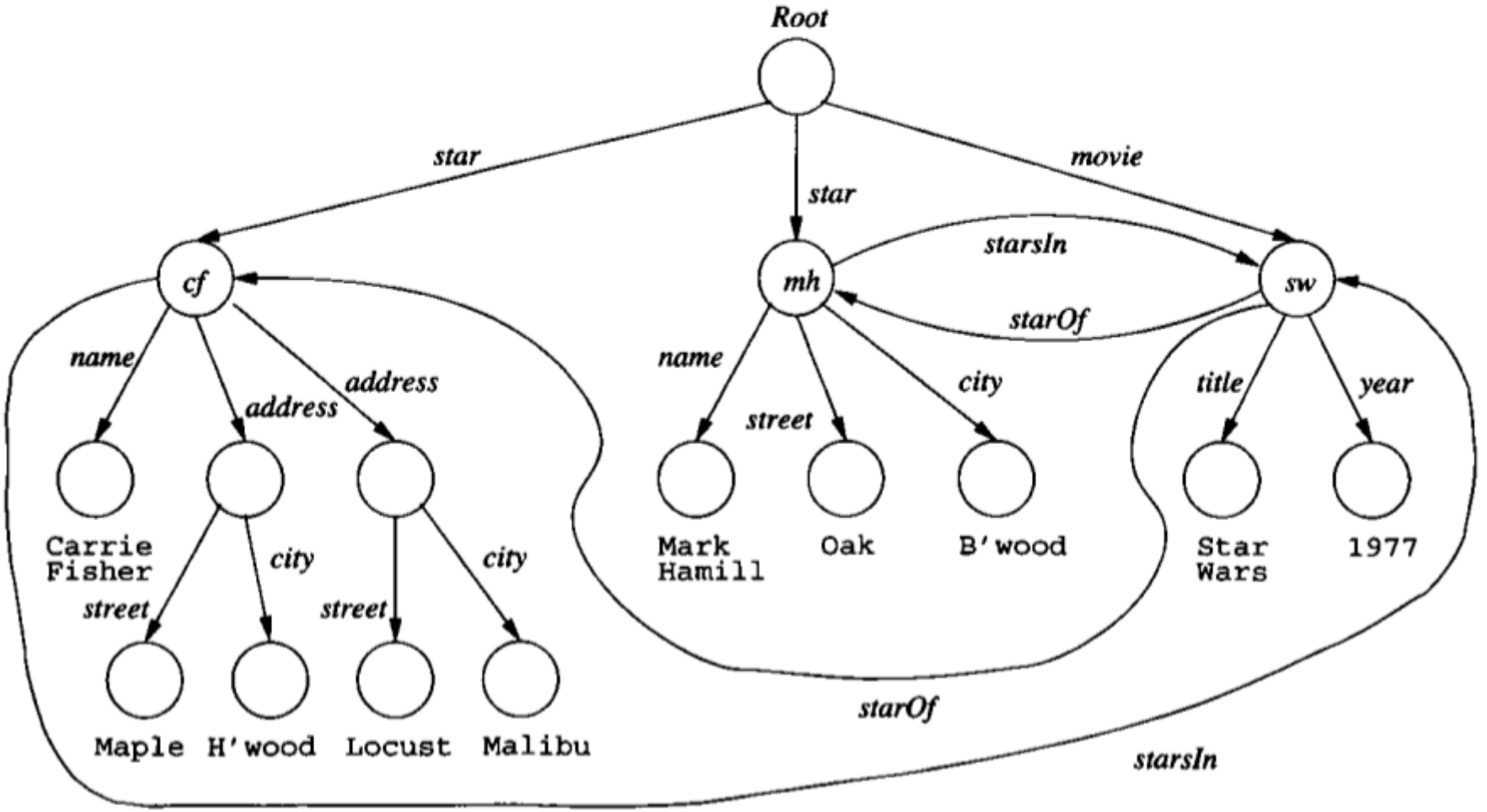
Resumo das diferenças entre dados estruturados e dados semiestruturados

Dados estruturados	Dados semiestruturados
Esquema pré-definido	Nem sempre há um esquema pré-definido
Estrutura regular	Estrutura irregular (em nível tanto de atributos quanto de tipos)
Estrutura independente dos dados	Estrutura embutida nos dados
Estrutura reduzida	Estrutura extensa (para refletir as particularidades de cada dado, já que cada dado pode ter uma organização própria)
Estrutura fracamente evolutiva	Estrutura fortemente evolutiva (a estrutura dos dados modifica-se tão frequentemente quanto os seus valores)
Estrutura prescritiva (que permite definir esquemas fechados e restrições de integridades com relação à semântica dos atributos)	Estrutura descritiva
Distinção entre estrutura e dado é clara	Distinção entre estrutura e dado não é clara

Modelagem de dados semiestruturados

- Modelos de dados para BDs Relacionais não são os mais apropriados para a representação de dados semiestruturados – neles, as ocorrências de dados devem apresentar uma mesma estrutura
- Modelos para dados semiestruturados são mais flexíveis – suportam representações heterogêneas de dados semanticamente iguais
- Geralmente, os modelos propostos para dados semiestruturados representam os dados como algum tipo de **grafo direcionado rotulado**

Modelo de grafo de dados semiestruturados



Exemplo de representação de dados semiestruturados por meio de grafo (dados de um filme e duas de suas estrelas)
Extraído de [3].

Modelo de grafo de dados semiestruturados

- **Nós do grafo**
  - **Internos:** objetos compostos
  - **Folhas:** têm algum valor atômico associado (como números ou strings)
- **Arcos –** relacionamentos *objeto* → *subobjeto* ou *objeto* → *valor*
- **Rótulo** em um arco – indica como o objeto de origem do arco se relaciona com o objeto destino
- Não há restrições com relação ao número de arcos que partem de um objeto origem – cada ocorrência de dado pode ter uma estrutura diferente
- Todo BD semiestruturado tem um **objeto raiz**, que é ponto de partida para a investigação da sua estrutura

Modelo de grafo de dados semiestruturados

- Os rótulos dos arcos desempenham dois papéis. Suponha que temos um arco rotulado com *L* do nó *N* para o nó *M*:
  - É possível pensar que N representa um objeto ou estrutura, e que M é um atributo do objeto ou um campo da estrutura. Assim, L representa o nome de um atributo ou campo, respectivamente.
    - Ex: *name* e *address* para *star*
  - Também é possível pensar que N e M são objetos, e que L é o nome de um relacionamento de N para M.
    - Ex: *starIn* e *startOf* entre *star* e *movie*

Object Exchange Model (OEM)

- Foi o primeiro modelo de dados proposto para dados semiestruturados
- Foi apresentado em 1995, como resultado de um projeto de integração de bancos de dados heterogêneos chamado ***Tsimmis***, da Universidade de Stanford
- No OEM, cada objeto é descrito por uma quádrupla

  <rótulo, tipo, valor, OID>

  - rótulo é uma string que descreve o que o objeto representa
  - tipo pode ser um tipo atômico ou um conjunto de objetos
  - valor é um campo que mantém um valor compatível com o tipo definido
  - OID é a identificação única do objeto e pode ser null

Object Exchange Model (OEM)

- O OEM é **autoexplicativo** – cada objeto possui um rótulo (~ nome de coluna no modelo relacional) e um tipo (~ tipo de coluna no modelo relacional)
- Exemplos:

  <universidade, string, “USP”>

  <instituto, string, “Instituto de Matemática e Estatística”>

  <departamento, string, “Departamento de Ciência da Computação”>

Object Exchange Model (OEM)

- Objetos podem ser **atômicos** (i.e., com tipos pré-definidos como string, integer, gif) ou **complexos**
- Objetos complexos são decompostos hierarquicamente em objetos menores (por meio de referências a outros objetos)
- Exemplos:

<nomeAutor, set, {prenome1, sobrenome1}>

prenome1: <prenome, string, “Richard”>

sobrenome1: <sobrenome, string, “Feynman”>

Object Exchange Model (OEM)

- Outro exemplo:

<listaLivros, set, {livro1, livro2, livro3}>

livro1: <livro, set,{autor1, autor2, título1, anoPublicacao1}>

livro2: <livro, set,{autor2, título2, anoPublicacao2}>

livro3: <livro, set,{autor3, título3, anoPublicacao3}>

autor3: <autor, set, {prenome3, sobrenome3}>

titulo3: <título, string, “Introduction to Database Systems”>

anoPublicacao3: <anoPublicacao, integer, 1980>

Object Exchange Model (OEM)

- O OEM é considerado um modelo para **instâncias de dados** semiestruturados porque representa valores de dados e os associa a rótulos, que descrevem o seu significado
- Existe uma extensão baseada em OEM para representar **apenas esquemas de dados semiestruturados**, os chamados ***data guides***
- Um ***data guide*** é um conceito **similar ao de metadado** em um BD Relacional

Object Exchange Model (OEM)

- A estrutura de qualquer objeto de um conjunto de dados semiestruturados precisa estar representada no *data guide*
- E o *data guide* só deve manter estruturas que existam no conjunto de objetos
- Um *data guide* funciona como um **esquema *a posteriori* dos dados**
- Os data guides auxiliam a formulação e o processamento de consultas

Linguagens de consulta para dados semiestruturados

- As linguagens para a consulta de dados semiestruturados podem oferecer dois tipos de expressões:
  - as que permitem que os usuários extraiam itens de uma instância de dados semiestruturados; ou
  - as que permitem que os usuários transformem uma instância em outra instância de dados semiestruturados

Linguagens de consulta para dados semiestruturados

- As linguagens podem ser classificadas sob diferentes perspectivas:
  - **Expressividade**: quais tipos de consultas ou transformações elas podem expressar?
  - **Consulta X reestruturação**: a linguagem de consulta possibilita apenas a extração de itens a partir dos dados ou permite também que os dados sejam transformados?
  - **Composicionalidade**: o resultado de uma consulta pode ser usado como entrada para um outra consulta expressa na mesma linguagem?

Linguagens de consulta para dados semiestruturados

- Linguagens que só fazem extração de dados não são composicionais, porque o resultado de suas consultas não são instâncias de dados semiestruturados
- Linguagens de transformação podem não ser composicionais se a composição de duas consultas não puder ser expressa na mesma linguagem

E na Sequência...

- XML
- JSON

Referências Bibliográficas

- [1] “Desmistificando XML: da Pesquisa à Prática Industrial”, Mirella M. Moro, Vanessa Braganholo. Em: André C. P. L. F. de Carvalho; Tomasz Kowaltowski (Editores) Atualizações em Informática 2009.
- [2] “Dados Semi-Estruturados”, Ronaldo dos Santos, Carina Friedrich Dorneles, Adrovane Kade, Carlos Alberto Heuser. [Material de um tutorial para o SBBD 2000].
- [3] “Sistemas de Bancos de Dados” (6ª edição), Elmasri e Navathe, Capítulo 12 - “XML – Extensible Markup Language”
- [4] “Databases Systems – The Complete Book”, Garcia-Molina, Ullman, Widom, Seções 4.6 e 4.7