Zusammenfassungen semistrukturierter Datenbestände

Querysprachen f. semistrukturierte Daten

- Mit OEM geht Querysprache einher.
- Diese Querysprache ähnelt OQL, erlaubt insbesondere Pfadausdrücke.
- Beispiele:
  - `select Restaurant.Entrée`
  - `select Restaurant.Name`
    `where Restaurant.Entrée = "Burger"`
- Schemainformation hilfreich für
  - Queryoptimierung,
  - Datentransformation.

Data Guides

- Data Guides: konkrete Zusammenfassung des Datenbank-Inhalts, wenn kein Schema vorhanden.
- Unterschied zwischen ‘DataGuide’ und ‘Schema’: DataGuide ist konform zur Datenbank, nicht umgekehrt.

DataGuides und Alternativen

- DataGuides: Relativ akkurate Beschreibung,
- k-ROs und k-Indices/T-Indices: ungefähre Beschreibung.

DataGuides

Kurze, akkurate, und ‘geeignete’ Zusammenfassung der Struktur einer Datenbank.

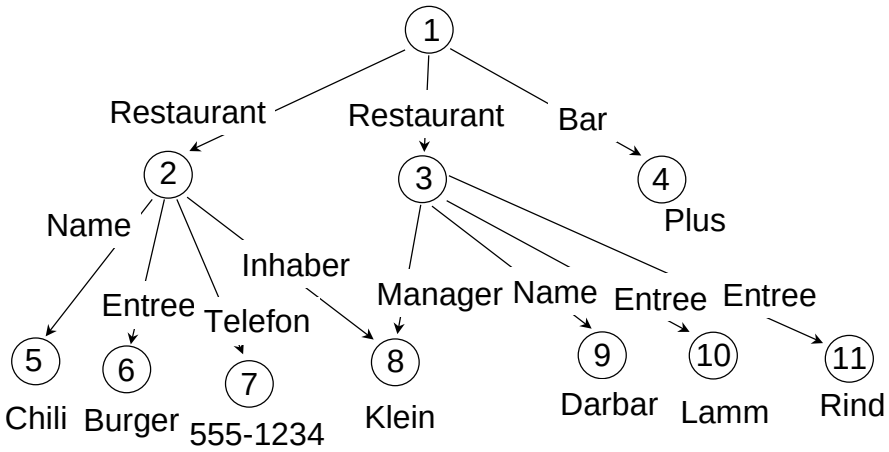
- Kürze: DataGuide beschreibt jeden label path mit einer Instanz in der Datenbank genau einmal.
- Akkuratheit: DataGuide beschreibt keine label paths, die nicht in der Datenbank vorkommen.
- ‘Geeignetheit’: DataGuide ist OEM Objekt (⇒ Speicherung und Zugriff auf DataGuides mit OEM-Mechanismen möglich.)

DataGuides - Definition

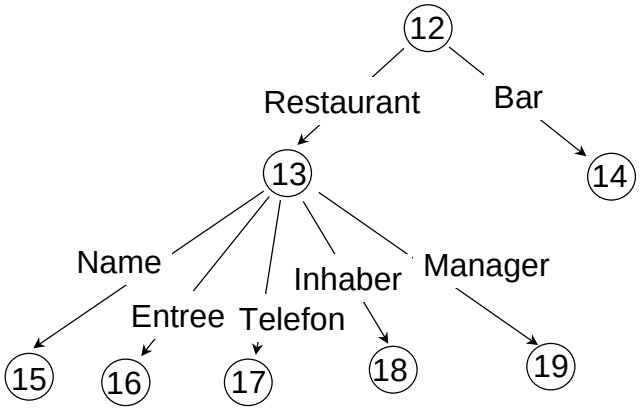
Ein DataGuide einer OEM-Datenbank s ist ein OEM Objekt d, so dass

- jeder label path in s genau eine data path-Instanz in d hat,
- jeder label path von d ein label path von s ist.

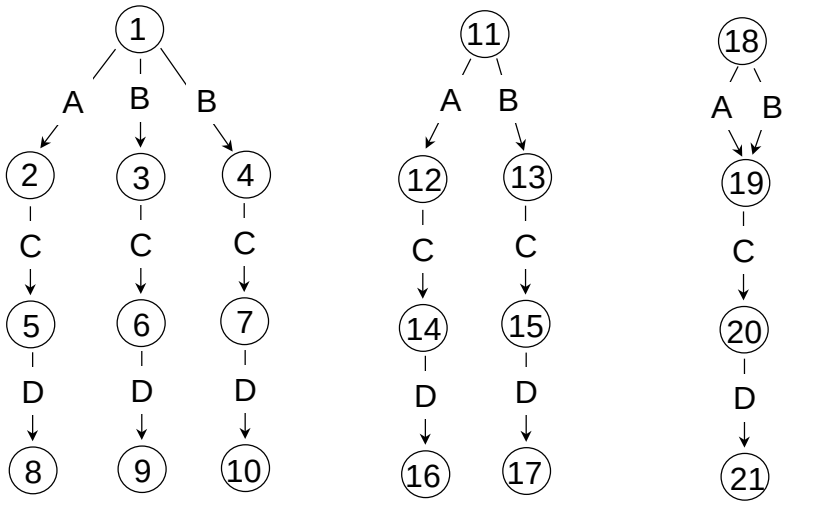
Beispiel-Datenbank



DataGuides - Beispiel



DataGuides - weiteres Beispiel



Datenbank

Zwei entsprechende DataGuides

Einordnung DataGuides

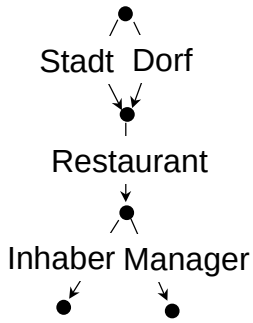
Noch akkuratere Beschreibung der Datenbank grundsätzlich möglich, z.B. um festzulegen, welche Kombination von Labels von ausgehenden Kanten vorkommen, z.B. ‘Inhaber’ oder ‘Manager’ (geht mit XML-DTDs).

Erzeugung von DataGuides

Äquivalent zu NEA -> DEA

Übung

- Ist der Graph rechts ein DataGuide, wenn Stadt-Restaurants nur Manager und Dorf-Restaurants nur Inhaber haben?
- Wie ist es, wenn ein Stadt-Restaurant beides hat, Dorf-Restaurants haben nur Inhaber?



Minimale DataGuides

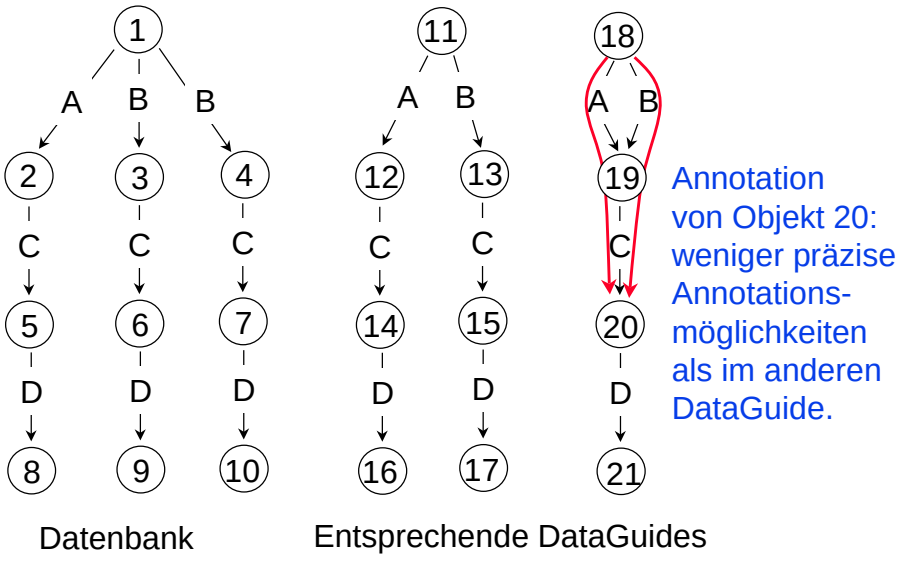
- Es existieren wohlbekannte Techniken zur Minimierung von DataGuides, d.h. zur Erzeugung eines DataGuides mit minimaler Anzahl von Zuständen aus beliebigem DataGuide.
- Nachteile minimaler DataGuides:
  - Änderungen an der Datenbank verursachen mehr Arbeit, Beispiel
  - *Annotationen*, d.h. Aussagen über Menge von Objekten, die über einen label path erreichbar sind, sind weniger gut möglich. Annotation f. Objekt 20

Strong DataGuides

- Motivation: Charakterisierung der DataGuides, deren Annotationen stets eindeutig sind.
- Intuition: Label paths mit dem gleichen (singleton) Target Set im DataGuide haben stets das gleiche Target Set in der Datenbank.

Naechste Folie Illustration.

Strong DataGuides - Illustration



Annotation von Objekt 20: weniger präzise Annotations-möglichkeiten als im anderen DataGuide.

Datenbank

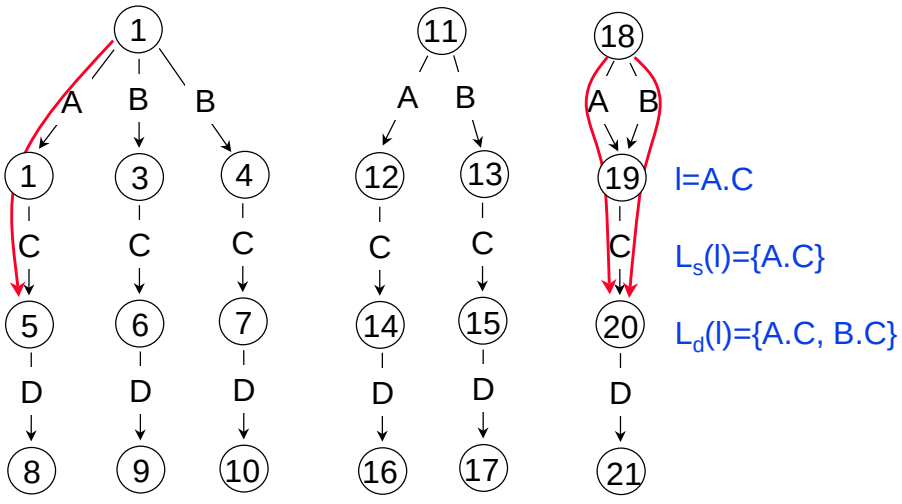
Entsprechende DataGuides

Strong DataGuides - Definition

- OEM Objekte s und d, d ist DataGuide für s,
- $T_s(l)$ - Target Set von l in s,
- $T_d(l)$ - (singleton) Target Set von l in d,
- $L_s(l) = \{m | T_s(m)=T_s(l)\}$,
  d.h. $L_s(l)$ ist die Menge aller label paths mit dem gleichen Target Set wie l,
- $L_d(l) = \{m | T_d(m)=T_d(l)\}$,
  d.h. $L_d(l)$ ist die Menge aller label paths in d mit dem gleichen Target Set wie l.
- d ist ein *Strong DataGuide*,
  wenn für alle label paths l von s: $L_s(l)=L_d(l)$

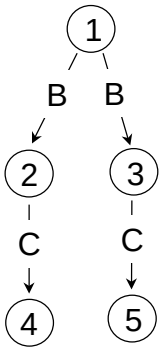
Am Beispiel erlaeutern - naechste Folie.

Strong DataGuides - Illustration



Datenbank Entsprechende DataGuides

Aufbau eines Strong DataGuides - Illustration



dg = 6 Neues Objekt
targetHash = {({1}, 6)} Hash-Tabelle
Aufruf ‘RecursiveMake({1}, 6)’

p={(B,2), (B,3)} Menge der Kinder eines der Objekte
l=B, t2={2,3}, d2=NIL
d2=7, targetHash = {({1}, 6), ({2,3}, 7)}
Aufruf ‘RecursiveMake({2,3}, 7)’

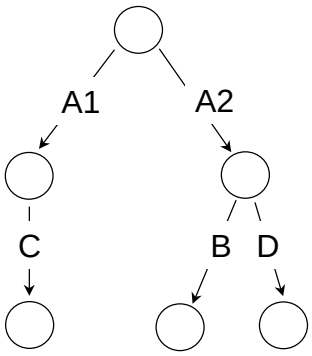
p={(C,4), (C,5)}
l=C, t2={4,5}, d2=NIL
d2=8
targetHash = {({1},6), ({2,3},7), ({4,5},8)}
Aufruf ‘RecursiveMake ({4,5},8)’

Query Processing mit DataGuides

- Aus dem DataGuide kann man für manche (Teil-)Queries ableiten, ob sie keine Lösung haben.
- Wenn man Target Sets beim DataGuide explizit abspeichert: Man erspart sich u.U. Navigieren im Datenbestand.
  Beispiel: Mit/ohne Index über Year-Werte.
  select DBS.Group_Member.Publication.Year
  where DBS.Group_Member.Publication.Year < 1975

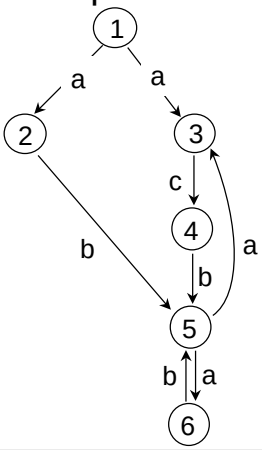
Query Processing mit DataGuides (2)

- Beispiel: gpe = Guide.A%.B%

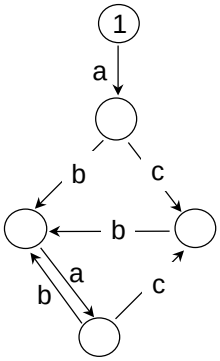


k-Representative Objects

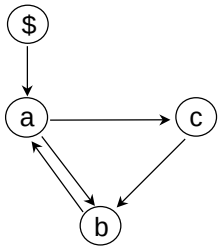
- Ziel: ‘Weniger ausführliche’ Beschreibung der Daten, die vorkommen dürfen.
- Beispiel:



DataGuide:



Anvisierte Struktur:



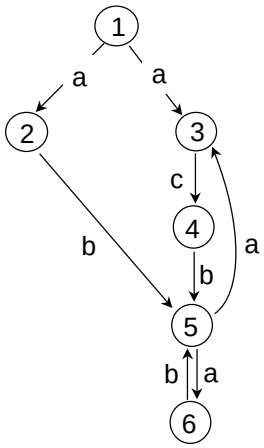
Labels an Knoten statt Kanten
k=1
$ - kuerzere Pfade - per Def. nur die direkt von der Wurzel.

k-Representative Objects

- k-Representative Object (k-RO) enthält
  die Pfade in der Datenbank bis zur Länge k+1.
- k-RO enthält
  Obermenge der Label Paths in der Datenbank.
- Anwendung:
  - Queryoptimierung,
  - gezielte Evaluierung von Pfadausdrücken.

Unterschied zu Optim. mit DataGuides erklaeren

Konstruktion k-Representative Objects



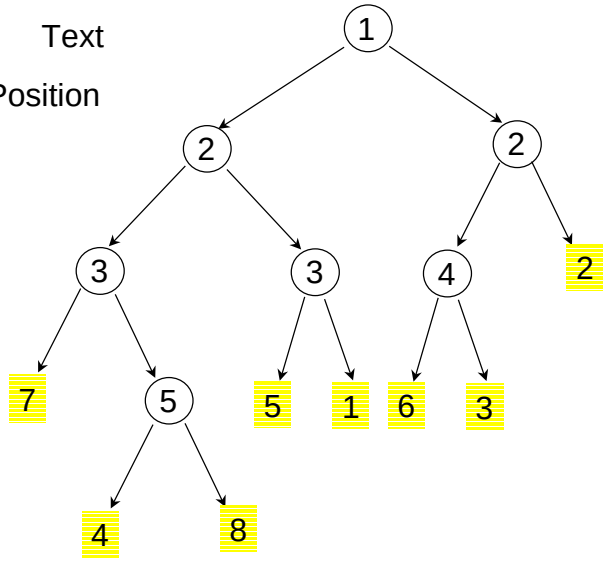
- k=1
  - Tripel konstruieren,
    (1 a 2), (1 a 3), …
  - Join-Verknüpfung
- k=2
  aba, acb, …

PAT-Trees

01100100010111… Text

123456789… Position

Warum folgt (5) auf (3)? Erklaeren

- Prefix Search
- Range Search
- regex Search

Automat auf Baum laufenlassen:
Zielzustand - Baum akzeptieren,
Blatt - Rest des Automaten laufenlassen
Beispiel
regex entspricht PAT-Anfrage