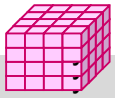
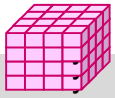
4. ETL: Schemaintegration + Data Cleaning

- ETL-Überblick
- Schemaintegration
- Schema Matching
  - Verfahren
  - Prototypen / Tools
- Data Cleaning
  - Probleme
  - Teilaufgaben
- Objekt-Matching (Entity resolution)
  - Techniken
  - MS SQL-Server
  - Prototypen U Leipzig



ETL-Prozess

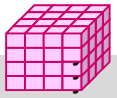
- Data Warehousing und ETL: materialisierter Ansatz zur Datenintegration
  - Erzeugung einer aggregierten, materialisierten Datenquelle für Online-Analysen
  - komplexer, aufwändiger Integrationsprozess
  - Offline-Durchführung erlaubt höhere Datenqualität gegenüber virtueller Datenintegration (Datentransformation während Query-Verarbeitung)
- *Extraktion*: Selektion eines Ausschnitts der Daten aus Quellen
  - ausgeführt an den entsprechenden Quellen
- *Transformation*: Aufbereitung und Anpassung der Daten an vorgegebene Schema- und Qualitätsanforderungen
  - ausgeführt im temporären Arbeitsbereich (Data Staging Area)
- *Laden*: physisches Einbringen der Daten aus Arbeitsbereich in das Data Warehouse, einschließlich evtl. notwendiger Aggregationen



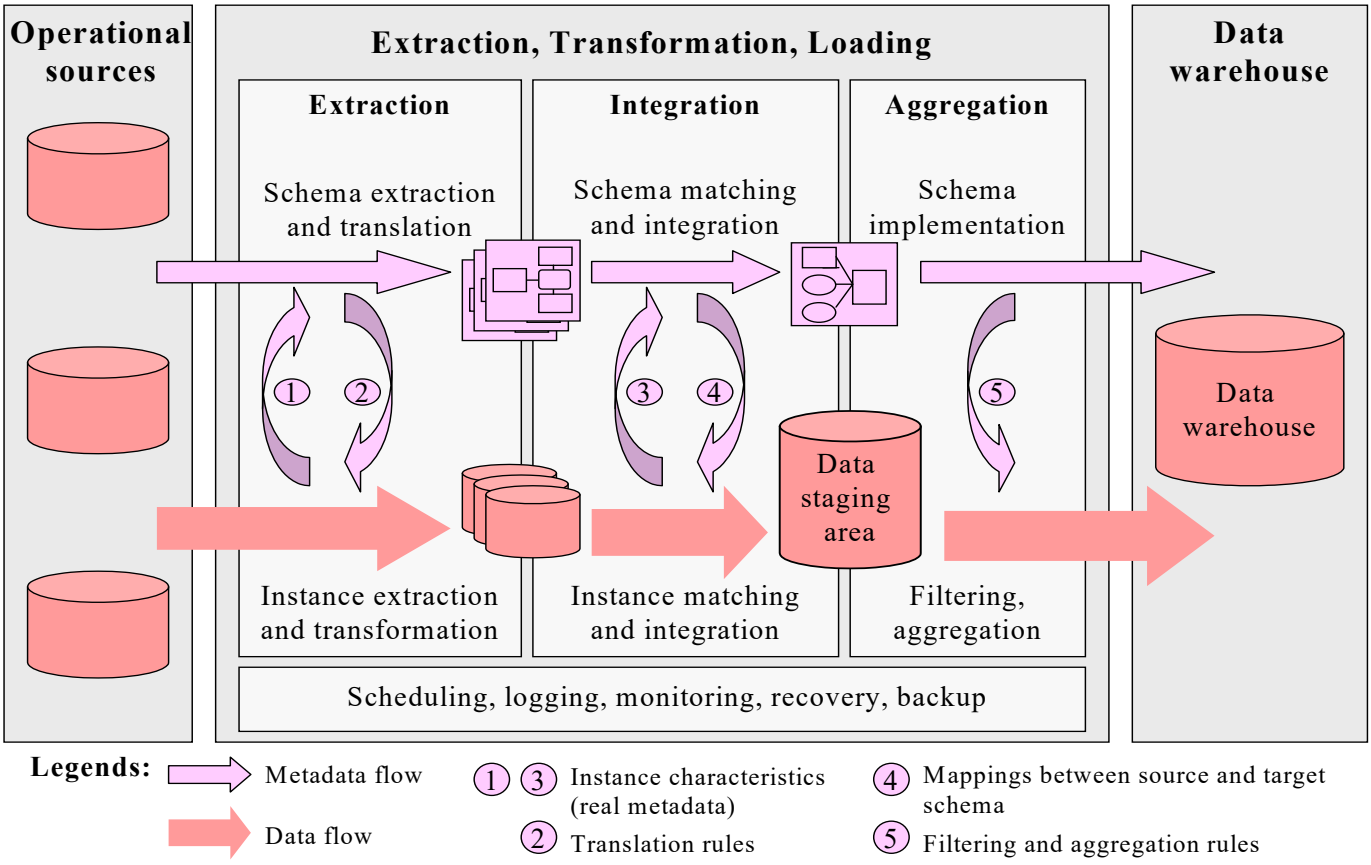
ETL-Prozess (2)

- Aufwändigster Teil des Data Warehousing
  - Vielzahl von operativen Quellen
  - Heterogenität der Datenquellen (DBMS, Schemata, Daten)
  - Gewährleistung hoher Qualität der Warehouse-Daten

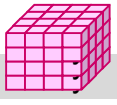
- Entscheidende Rolle im Data Warehousing, da großer Einfluss auf
  - Genauigkeit und Richtigkeit der später durchgeführten Analysen
  - die darauf basierenden Entscheidungen: „Garbage In, Garbage Out“



ETL-Prozess: Ablauf

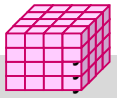


E. Rahm, H. H. Do: *Data Cleaning: Problems and Current Approaches.* IEEE Techn. Bulletin on Data Engineering, 2000



ETL als Integrationsprozess

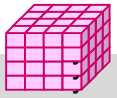
- ETL: Integration auf 2 Ebenen, Schemaintegration und Datenintegration
- Schemaintegration
  - Konstruktion eines Data Warehouse-Schemas aus existierenden Quellschemata
  - Ableitung von Korrespondenzen zwischen dem Data Warehouse-Schema und existierenden Quellschemata: *Schema Matching*
- Datenintegration / Data Cleaning
  - Transformation heterogener Daten in die einheitliche, durch das Data Warehouse-Schema vorgeschriebene Repräsentation
  - Entdeckung und Behebung von Datenqualitätsproblemen
  - Entdeckung äquivalenter Objekte/Sätze (Korrespondenzen auf Instanzenebene): *Objekt-Matching* / Duplikaterkennung



Schemaintegration - Anforderungen

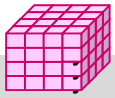
- Minimalität
  - keine Redundanz im integrierten Schema
  - Abbildung mehrerer gleicher/ähnlicher Konzepte in lokalen Schemata auf ein Konzept im integrierten Schema
- Korrektheit
  - Äquivalenz der im integrierten Schema enthaltenen Informationen mit denen in den lokalen Schemata
  - Konsistenz der während der Integration ergänzten Informationen, z.B. Beziehungen zwischen Konzepten im integrierten Schema
- Verständlichkeit

Vollständigkeit (Beibehaltung aller Informationen aus Quellschemas) ?



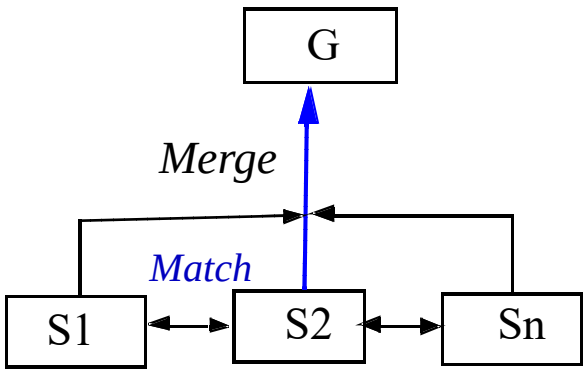
Schemaintegration (2)

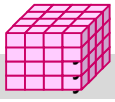
- Probleme der Schemaintegration
  - Heterogenität der Schemarepräsentationen, z.B. relational (SQL), XML, Entity-Relationship (ER), objekt-orientiert (UML), ...
  - semantische Heterogenität der Schemaelemente (Namenskonflikte, strukturelle Konflikte)
- Alternativen
  - Bottom-Up-Schemaintegration
  - Top-Down-Schemaintegration



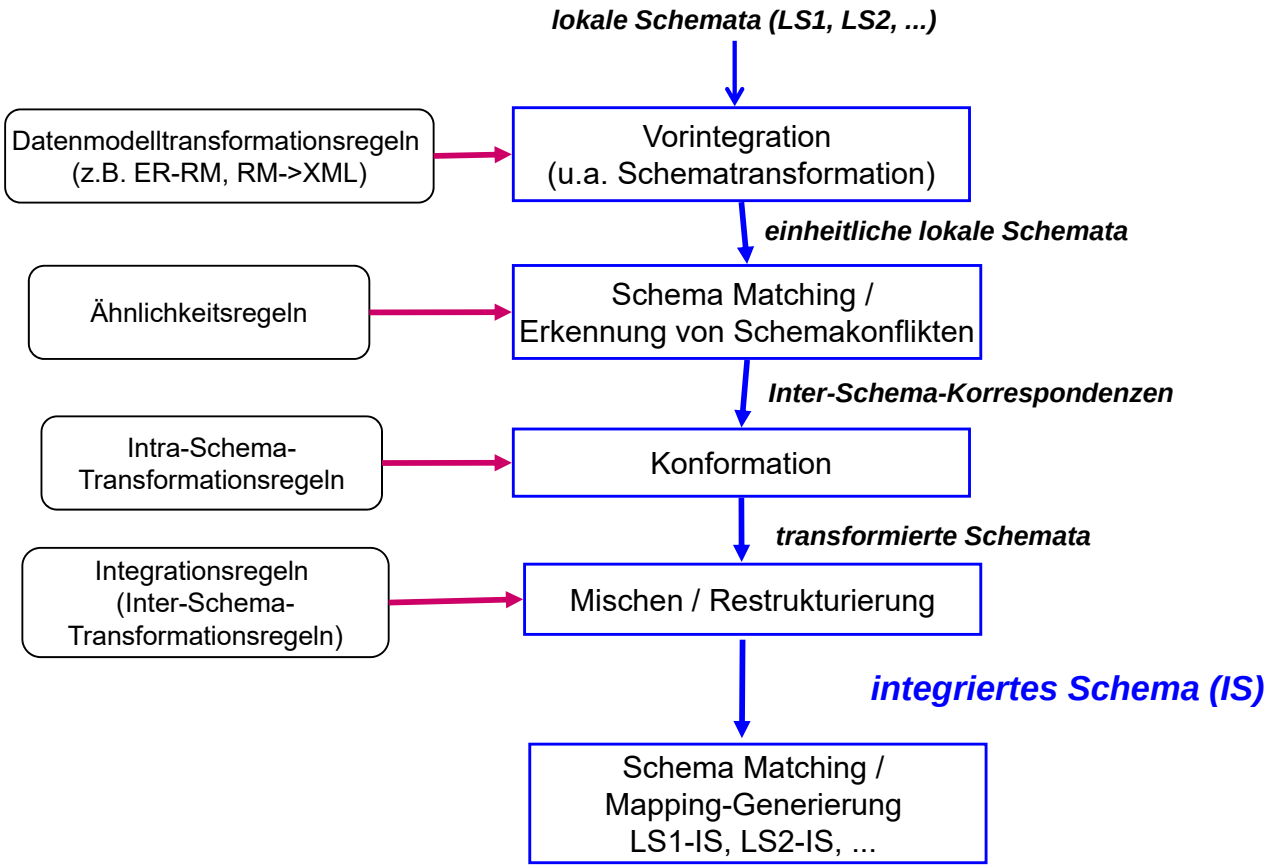
Bottom-Up-Integration (*Global as View*)

- (vollständiges) Mischen aller Source-Schemata in globales Schema
- setzt Abgleich zwischen Source-Schemas voraus, insbesondere Bestimmung von Korrespondenzen / Konflikten
- globales Schema entspricht gemeinsamer Sicht (View) auf die zugrundeliegenden Quellen
- neue Quelle ändert meist globales Schema



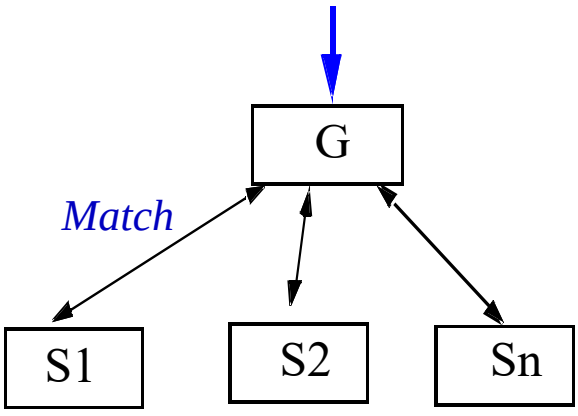


Bottom-Up-Schemaintegration (2)

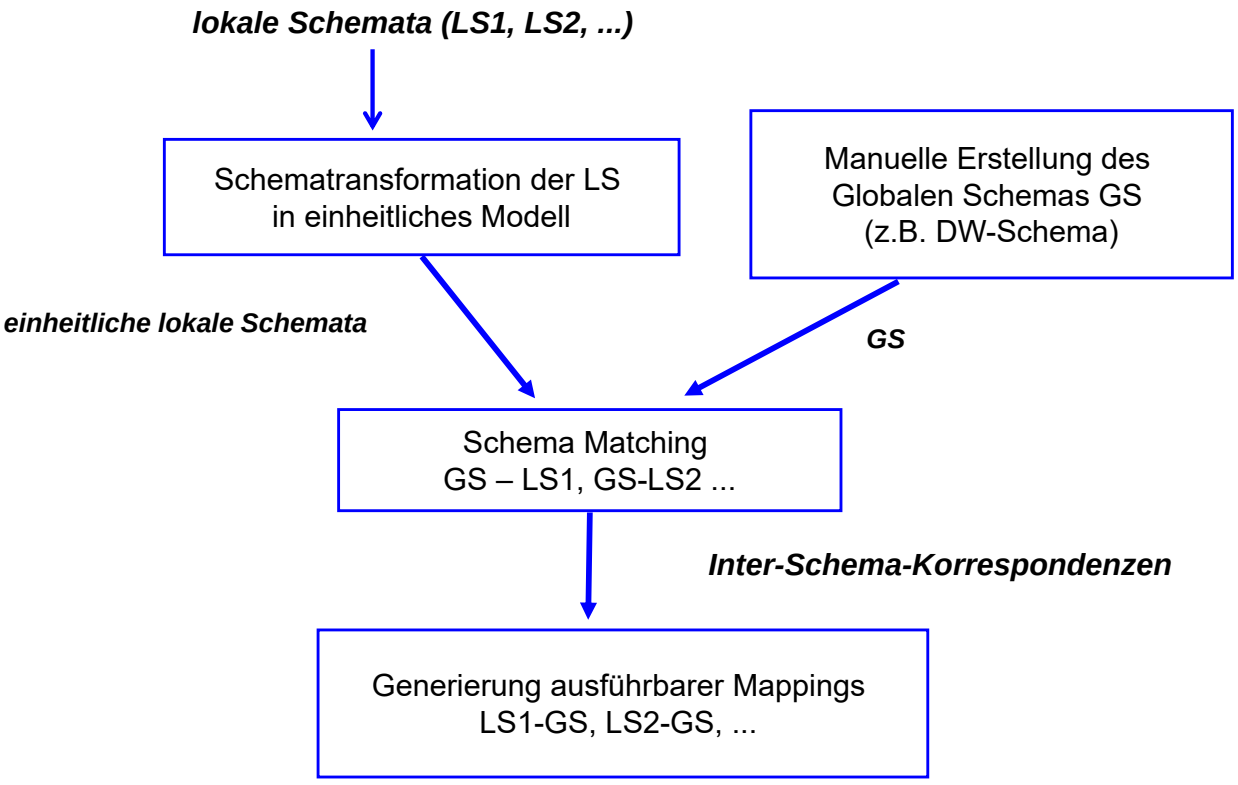


Top-Down-Integration (*Local as View*)

- globales Schema G ist vorgegeben
- jede Source S wird unabhängig von anderen Sources mit globalem Schema abgeglichen, d.h. ein Mapping G - S erstellt (Mapping beschreibt Inhalt der Quelle)
- aufwändige Query-Verarbeitung bei virtueller Integration
- G berücksichtigt i.a. nur Teile der lokalen Schemata



Top-Down-Schemaintegration (2)



Namenskonflikte

- **Synonyme**: Repräsentation ein und desselben Konzepts durch unterschiedliche Namen:
- **Homonyme**: Nutzung gleicher Namen für verschiedene Konzepte
- **Hyponyme/Hyperonyme:** Unter-/Oberbegriffe

Mitarbeiter
Name
Adresse

Firma
Name
Adresse

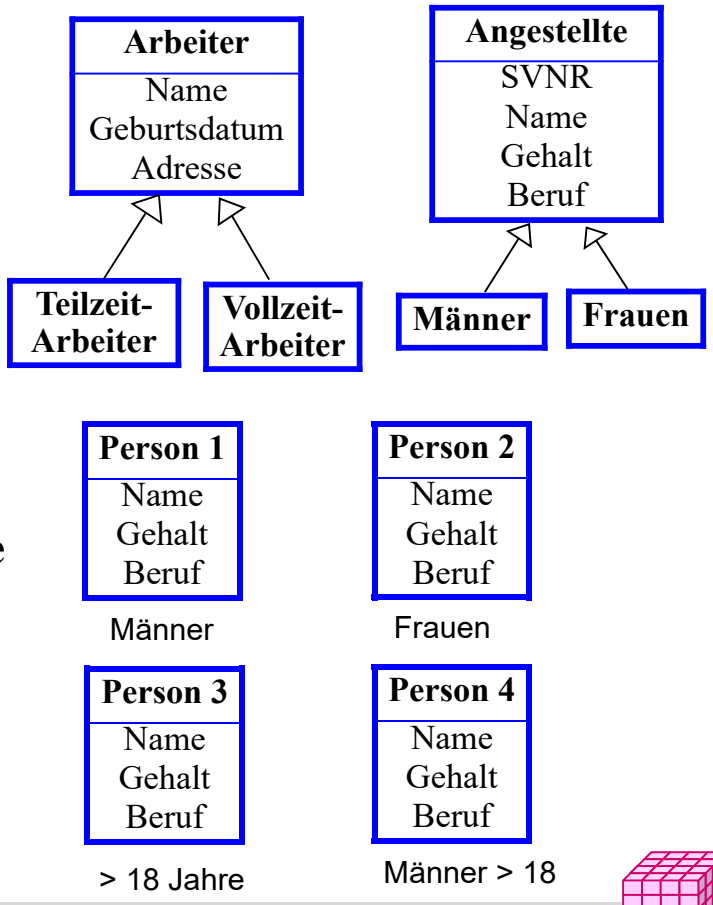
Angestellte
Name
Anschrift

Strukturelle Konflikte

- Entity vs. Entity
  - unterschiedliche Schlüssel
  - unterschiedliche Attributmengen, fehlende Attribute
  - unterschiedliche Abstraktionsebenen (Generalisierung, Aggregation)
  - **unterschiedliche Realitätsauschnitte** (RWS, real world states)
    disjunkt (disjoint):

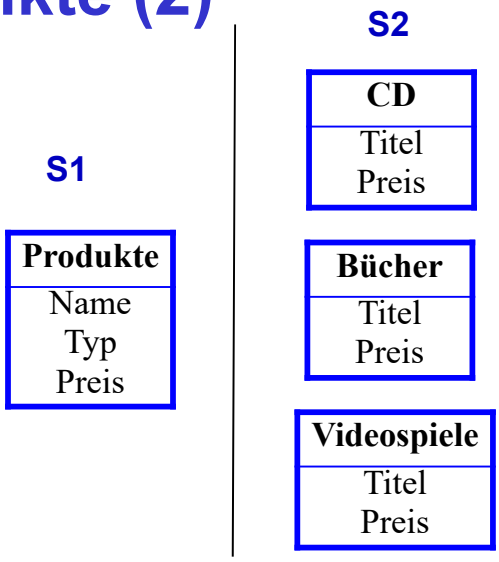
    überlappend (overlaps):

    enthalten (contains):



Strukturelle Konflikte (2)

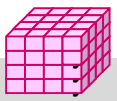
- Attribut vs. Entity-Konflikte
  - Repräsentation von Attributen als eigenständige Entities/Relationen



- Attribut vs. Attribut-Konflikte
  - unterschiedliche Datentypen
    Preis (Float) vs. Preis (String)
  - unterschiedliche Detailgrade
    Name vs. Vorname und Nachname
  - unterschiedliche Einheiten: *$ vs. Euro*
  - unterschiedliche Genauigkeiten: *Tausend Euro vs. Euro*
  - unterschiedliche Integritätsbedingungen, Wertebereiche, Default-Werte ...
    Alter >18 vs. Alter > 21
  - unterschiedliche Behandlung von Nullwerten
  - unterschiedliche Verwaltung der referentieller Integrität

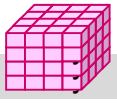
Behandlung von Konflikten

- Konflikterkennung: Vergleich der Schemata
  - Identifikation der ähnlichen/gleichen in den Schemata enthaltenen Information
  - Identifikation verschiedener Strukturen, die ähnliche Informationen repräsentieren
- Repräsentation der Inter-Schema-Korrespondenzen
  - Synonyme, Matches: Kunde = Klient
  - Is-A-Korrespondenzen: Angestellte is-a Person
  - RWS-Korrespondenzen: RWS(Produkte) contains RWS(Bücher)
- Behebung von Schemakonflikten v.a. bei Bottom-Up-Integration (Merge) erforderlich
  - Umbenennungen zur Behebung von Namenskonflikten
  - Schemakonformation und –restrukturierungen
  - Top-Down-Integration: Spezifikation notwendiger Transformationen bei Mapping-Erzeugung



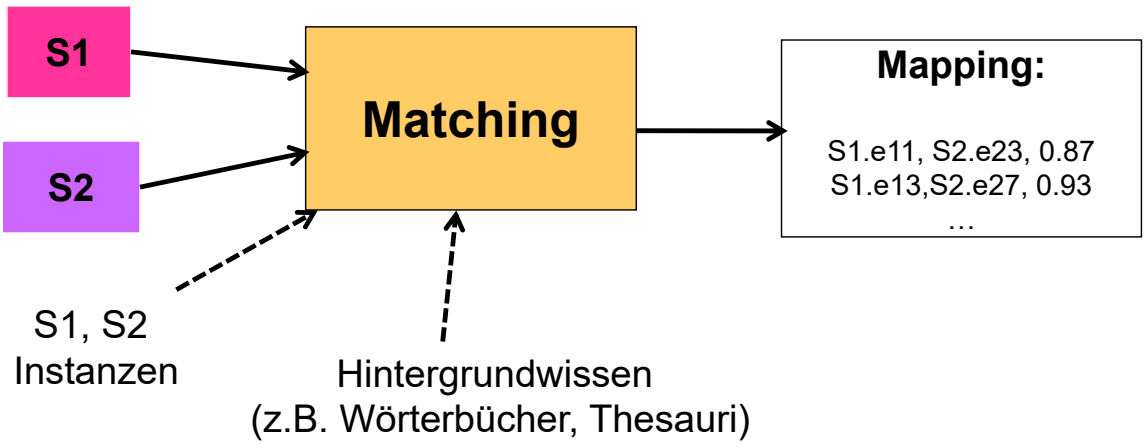
Automatisierungsbedarf

- bisherige Schemaintegrationsansätze weitgehend manuell
  - nutzerdefinierte Korrespondenzen und Konfliktbehandlung
  - Nutzung spezifischen Schema-/Domain-Wissens
  - aufwändig / fehleranfällig vor allem für größere Schemata
  - nicht skalierbar auf viele Schemata
  - Hoher Anpassungsaufwand bei Schemaänderungen
- Skalierbarkeit erfordert semi-automatische Lösungen / Tools!
  - vollautomatische Lösungen aufgrund semantischer Heterogenität nicht möglich
  - Namensproblematik (Synonyme, Homonyme)
  - begrenzte Mächtigkeit von Metadaten / Schemasprachen
- (Teil-)Automatisches Schema-Matching
  - v.a. für große Schemata wichtig
  - Nutzer-Feedback notwendig, jedoch im begrenzten Umfang

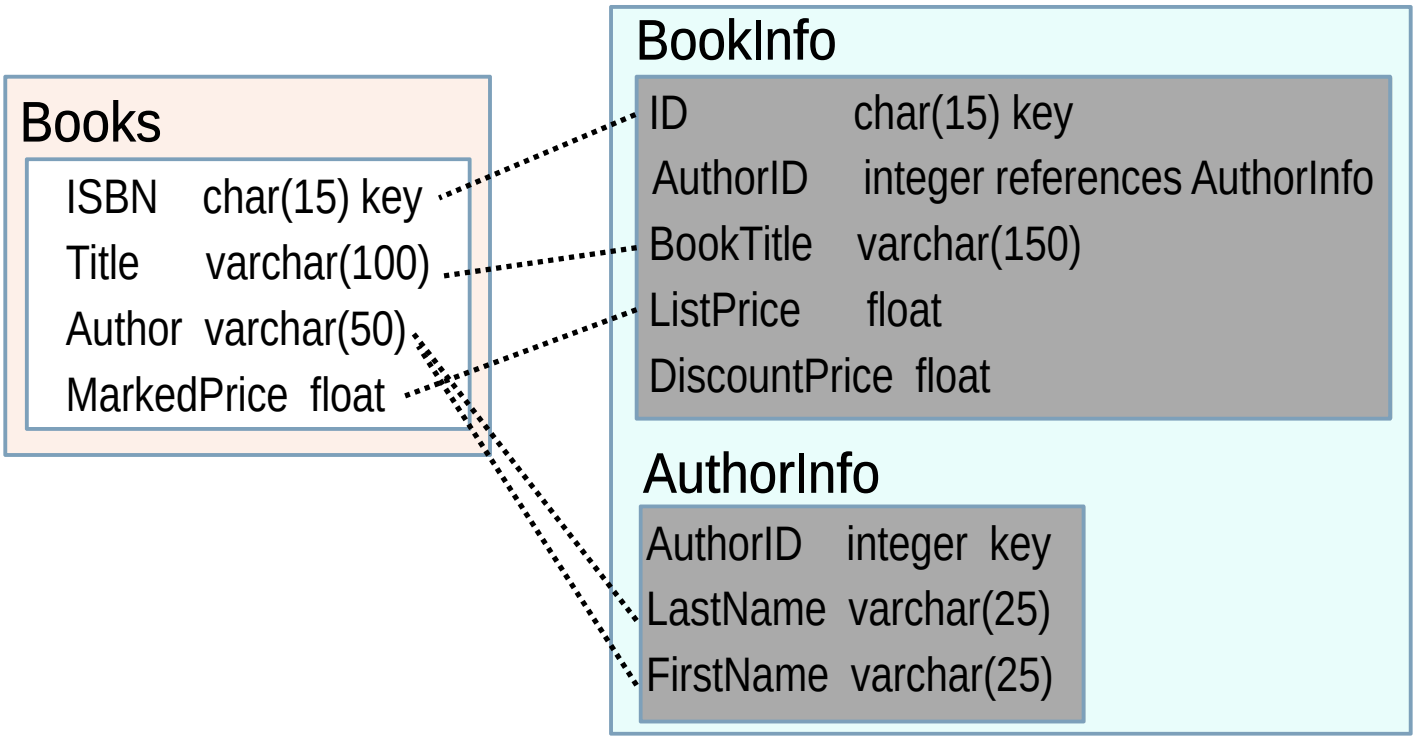


Schema Matching

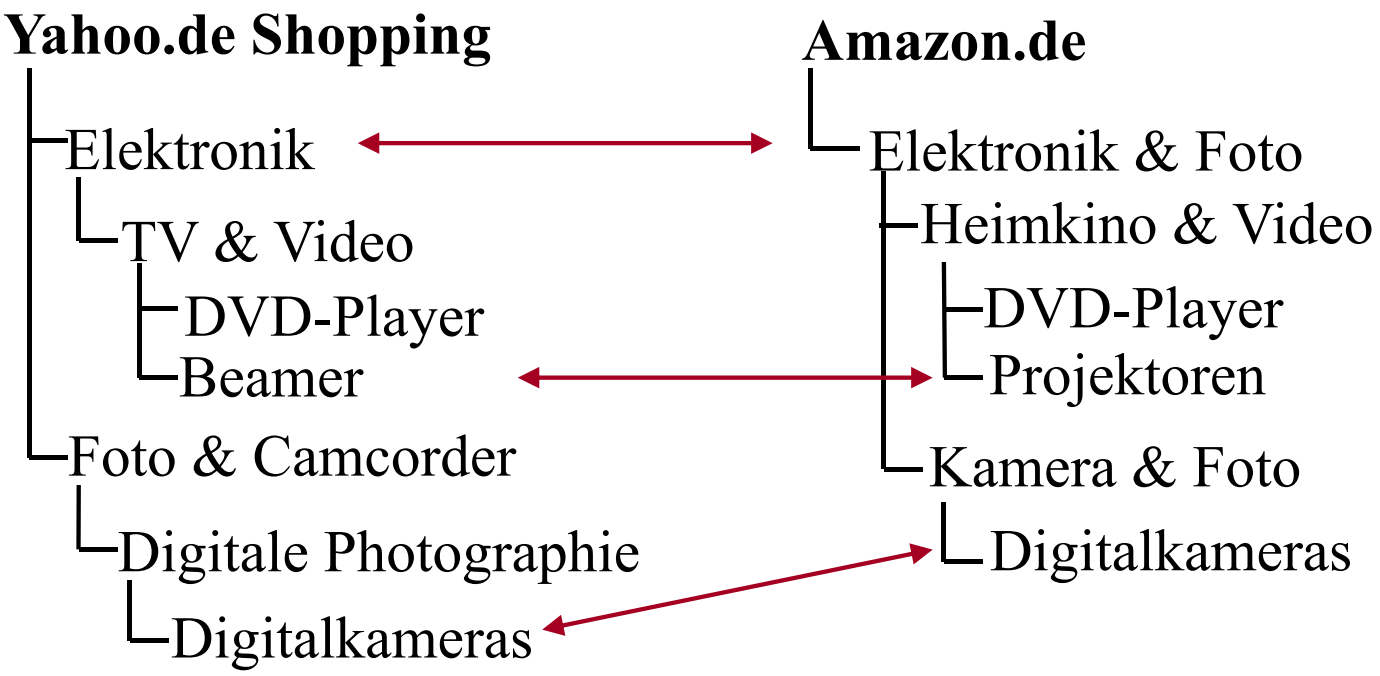
- Finden semantischer Korrespondenzen zwischen 2 Schemas
  - DB-Schemas, XML-Schemas, Ontologien, …
- Kritischer Schritt in zahlreichen Applikationen
  - Datenintegration: Data Warehouses, Mediatoren, P2P
  - E-Business: XML Message Mapping; Katalogintegration
  - Semantic Web: Ontology Matching



Match-Beispiel 1: relationale Schemas



Match-Beispiel 2: Produktkataloge



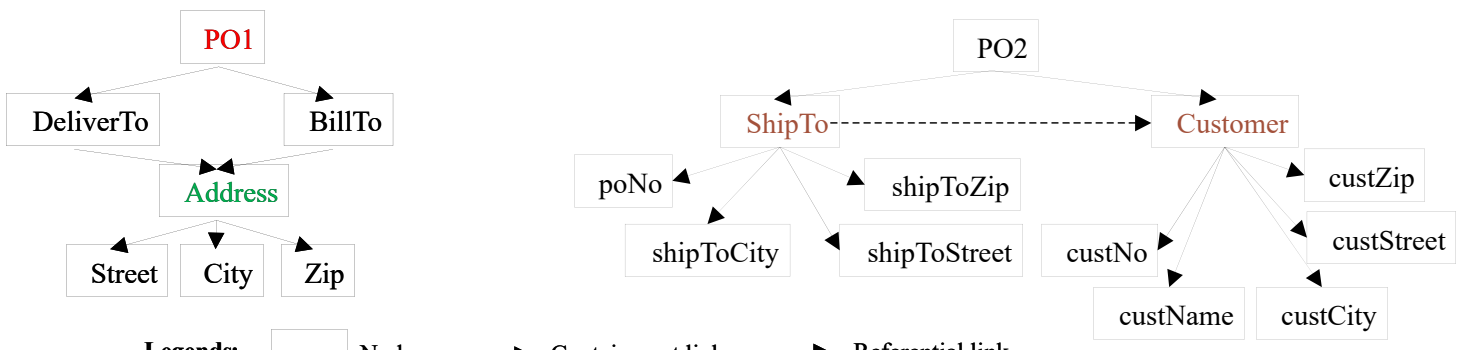
Match-Beispiel 3: Schemas (XML, relational)

```
<xsd: schema xmlns:xsd="http://www.w3.org/2001/ XMLSchema">
<xsd: complexType name ="PO1" >
  <xsd: sequence>
      <xsd: element name="DeliverTo" type="Address"/>
      <xsd: element name="BillTo" type="Address"/>
  </xsd: sequence>
</xsd: complexType>
<xsd: complexType name="Address" >
  <xsd: sequence>
      <xsd: element name="Street" type="xsd: string"/>
      <xsd: element name="City" type="xsd: string"/>
      <xsd: element name="Zip" type="xsd: decimal"/>
  </xsd: sequence>
</xsd: complexType>
</xsd: schema>
```

PO schema 1 (XML)

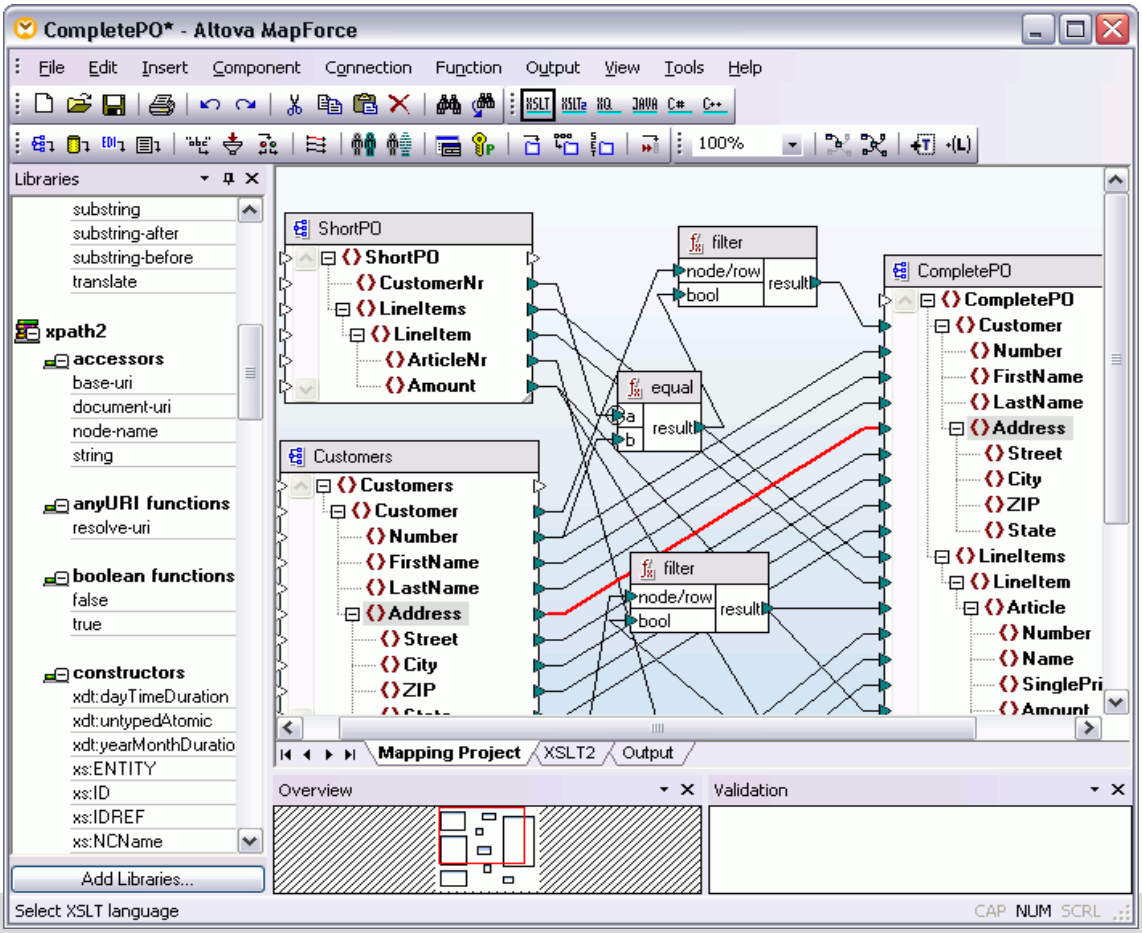
```
CREATE TABLE PO2. ShipTo (
 poNo          INT,
 custNo        INT REFERENCES PO2.Customer,
 shipToStreet  VARCHAR(200),
 shipToCity    VARCHAR(200),
 shipToZip     VARCHAR(20),
 PRIMARY KEY (poNo)
) ;
CREATE TABLE PO2. Customer (
 custNo        INT,
 custName      VARCHAR(200),
 custStreet    VARCHAR(200),
 custCity      VARCHAR(200),
 custZip       VARCHAR(20),
 PRIMARY KEY (custNo)
) ;
```

PO schema 2 (relational)



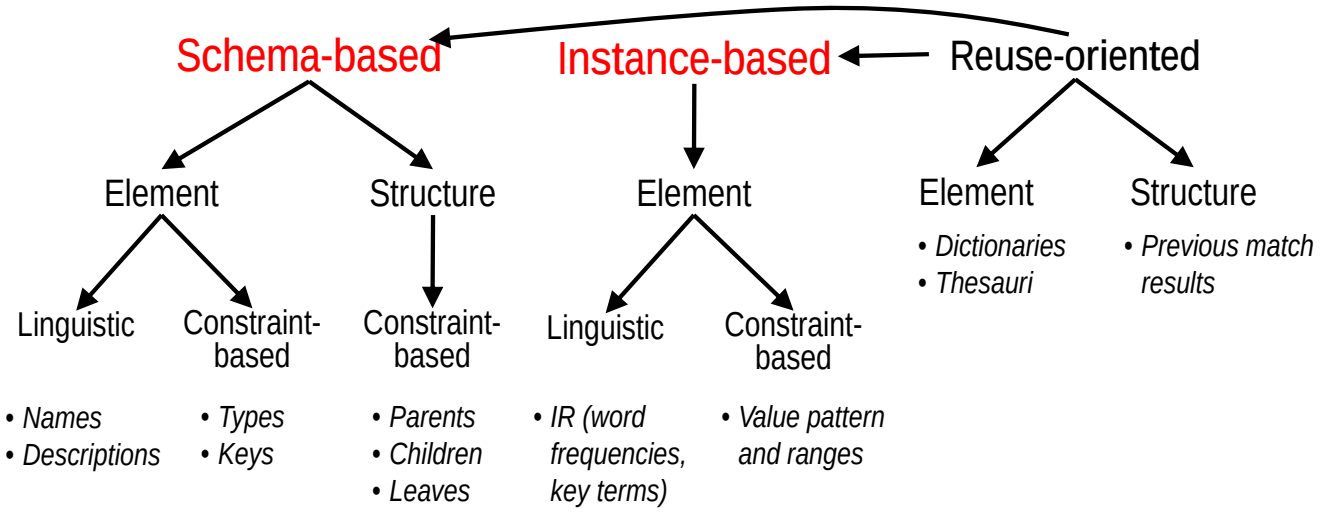
Graph representation of schemas (COMA++)

Match Tool: Altova MapForce



Automatische Match-Ansätze*

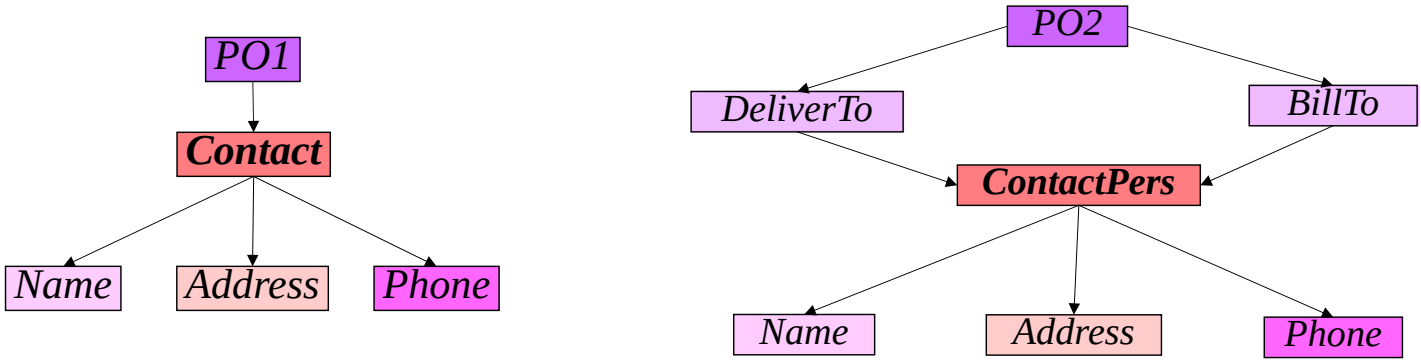
- einzelne Ansätze



- kombinierende Ansätze
  - Hybride Matcher (z.B. Name +Type)
  - Kombination unabhängiger Matcher (Match-Workflows)

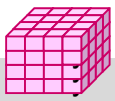
* Rahm, E., P.A. Bernstein: *A Survey of Approaches to Automatic Schema Matching.* VLDB Journal 10 (4), 2001

Match-Granularität



- **Element Matching**
  - Matching zwischen einzelnen Schema-Elementen (Blätter, innere Knoten)
  - Knoten- vs. Pfad-Repräsentation von Elementen

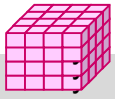
- **Strukturelles Matching**
  - Matching zwischen Teilschemas / Elementgruppen



Linguistisches Matching

- einfachster Ansatz: Namensgleichheit
- syntaktische Ansätze: approximate String-Ähnlichkeit von Namen
  - N-grams, Edit distance, TF/IDF ...
- semantische Ähnlichkeit durch Nutzung terminologischer Beziehungen
  - Synonyme: *KFZ* ~ *Auto*
  - Hyponyme/Hyperonyme: *Buch* is-a *Publikation*, *Artikel* is-a *Publikation*
  - Nutzung von Wörterbüchern / Thesauri, z.B. WordNet

- **Vorverarbeitung** zur Behandlung kryptischer Namen, Auflösung von Abkürzungen, etc.
  - Tokenisierung von Namen: *PO_OrderNum* → {*PO*, *Order*, *Num*}
  - Expansion von Akronymen, Kurzformen: *PO* →
    Num →

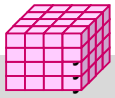


Ähnlichkeitsmaße für Strings: Edit-Distance

- Gegeben seien zwei Strings a und b (|a|=n, |b|=m)
- Editabstand d(a,b) = Anzahl der Edit-Operationen, um a in b zu konvertieren
  - Edit-Operationen: Einfügen, Löschen oder Ersetzen eines Zeichens
- Ähnlichkeit ist normierter Editabstand
  - auch: Levenshtein-Abstand

$$sim_{edit}(a,b) = 1 - \frac{dist(a,b)}{\max(|a|,|b|)}$$

- Beispiel: „Street“ vs. „ShipToStreet“



Ähnlichkeitsmaße für Strings: q-gram

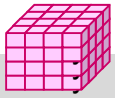
- Gegeben seien zwei Strings a und b (|a|=n, |b|=m)
- Idee: Wie viele gleiche Substrings der Länge q enthalten a und b?
  - Häufig q = 3 (Trigram)
- Ähnlichkeit mittels Dice-Koeffizient

$$sim_{qgram}(a,b) = \frac{2 \cdot |Q(a) \cap Q(b)|}{|Q(a)| + |Q(b)|}$$

  - Q(a) = Menge der q-Gramme von a
  - zu vergleichende Strings erhalten optional Präfix und Suffix mit je q-1 Füllzeichen (padding)

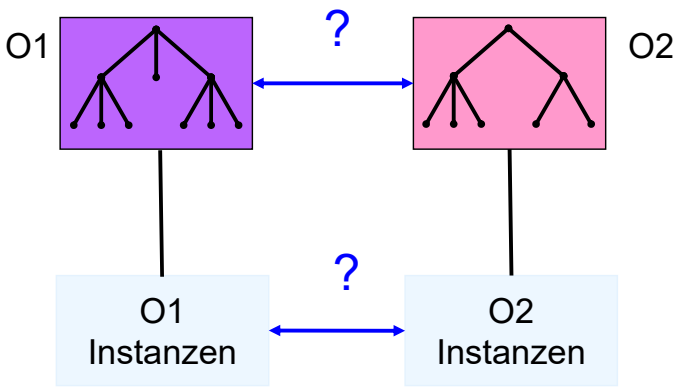
- Beispiel: „Street“ vs. „ShipToStreet“
  - ohne Padding

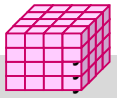
  - mit Padding:



Instanz-basiertes Schema Matching

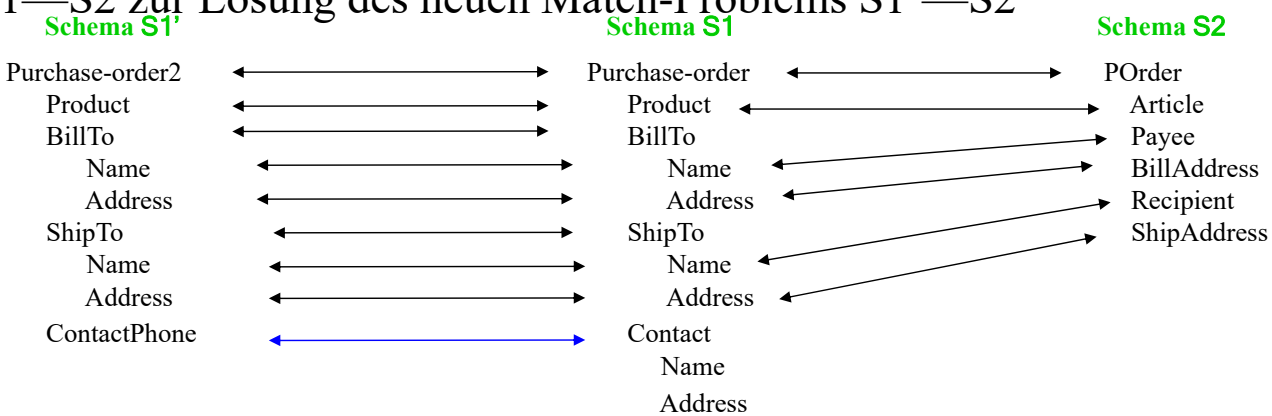
- Semantik von Schemaelementen / Ontologiekonzepten wird oft besser durch die zugeordneten Instanzdaten als durch Namen, Beschreibung etc. ausgedrückt
- Match für Elemente mit den ähnlichsten Instanzdaten
  - erfordert jedoch ähnliche Instanzdaten für viele Elemente/ Konzepte
- Überführung des Schema-Matching- in ein Objekt-Matching-Problem

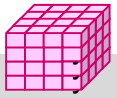




Wiederverwendung (Reuse)

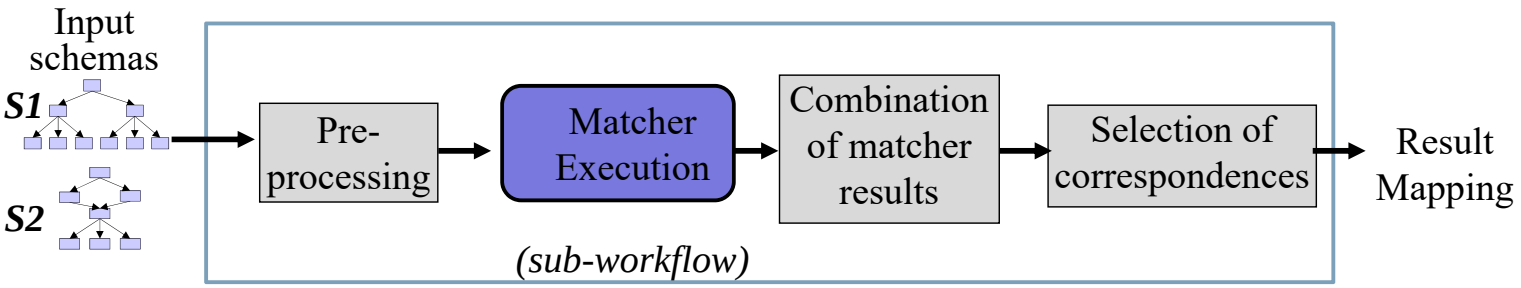
- Nutzung von Hilfsquellen
  - nutzerspezifizierte Synonymtabellen
  - allgemeine/domänenspezische Vokabulare, Wörterbücher
  - gemeinsame Ontologien
- Nutzung bereits bestätigter Match-Ergebnisse für ähnliche Match-Probleme
  - Speichern von Schemas und Mappings in Repository
  - besonders vorteilhaft für Abgleich neuer Schema-Versionen (Schema-Evolution)
- Beispiel: Wiederverwendung des vorhandenen (bestätigten) Mappings S1—S2 zur Lösung des neuen Match-Problems S1‘—S2



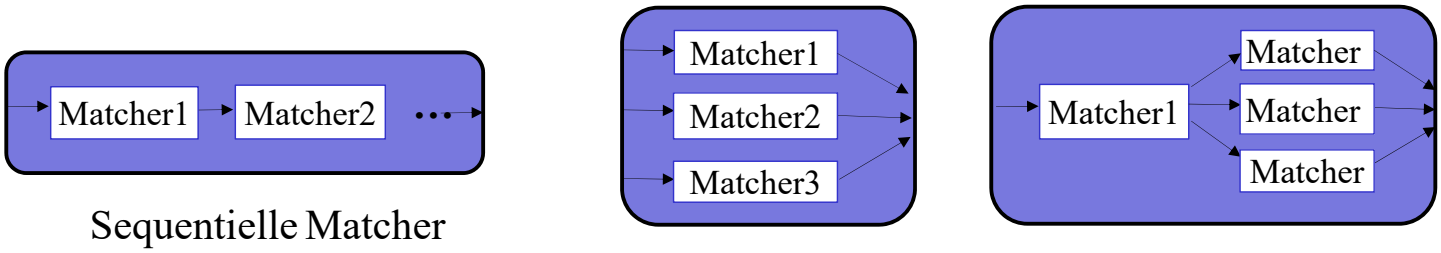


Kombination von Matchern

allgemeiner Match-Workflow (COMA, …)



Matcher-Ausführung:



Sequentielle Matcher

Parallele (unabhängige) Matcher

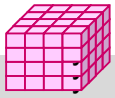
Gemischte Strategie

Match-Prototypen

Vergleich bekannter Match-Prototypen *

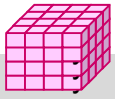
		Cupid	COMA++	Falcon	Rimom	Asmov	Agr.Maker	OII Harmony	GOMMA
Year of introduction		2001	2002/2005	2006	2006	2007	2007	2008	2010
Input schemas	*relational*	√	√	-	-	-	-	√	-
	XML	√	√	-	-	-	√	√	-
	ontologies	-	√	√	√	√	√	√	√
OAEI participation		-	√	√	√	√	√	-	√
Comprehensive GUI		-	√	(√)	?	?	√	√	-
Matchers	*linguistic*	√	√	√	√	√	√	√	√
	structure	√	√	√	√	√	√	√	√
	instance	-	√	-	√	√	√	-	√
Use of external dictionaries		√	√	?	√	√	√	√	(-)
Schema partitioning		-	√	√	-	-	-	-	√
Parallel matching		-	-	-	-	-	-	-	√
Dynamic matcher selection		-	-	-	√	-	-	-	-
Mapping reuse		-	√	-	-	-	-	-	√

*Rahm, E.: Towards large-scale schema and ontology matching. In: Schema Matching and Mapping, Springer-Verlag, 2011



Kommerzielle Match Tools

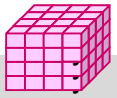
- viele GUI-basierte Mapping Editors zur manuellen Spezifikation von Korrespondenzen und Mappings
- zunehmende Unterstützung für semi-automatisches Matching, v.a. linguistisches Matching
  - Altova MapForce
  - MS BizTalk Server
  - SAP Netweaver
  - IBM Infosphere
- viele weitere Verbesserungen noch möglich
  - strukturelles / instanz-basiertes Matching
  - Techniken für große Schemas



Data Cleaning*

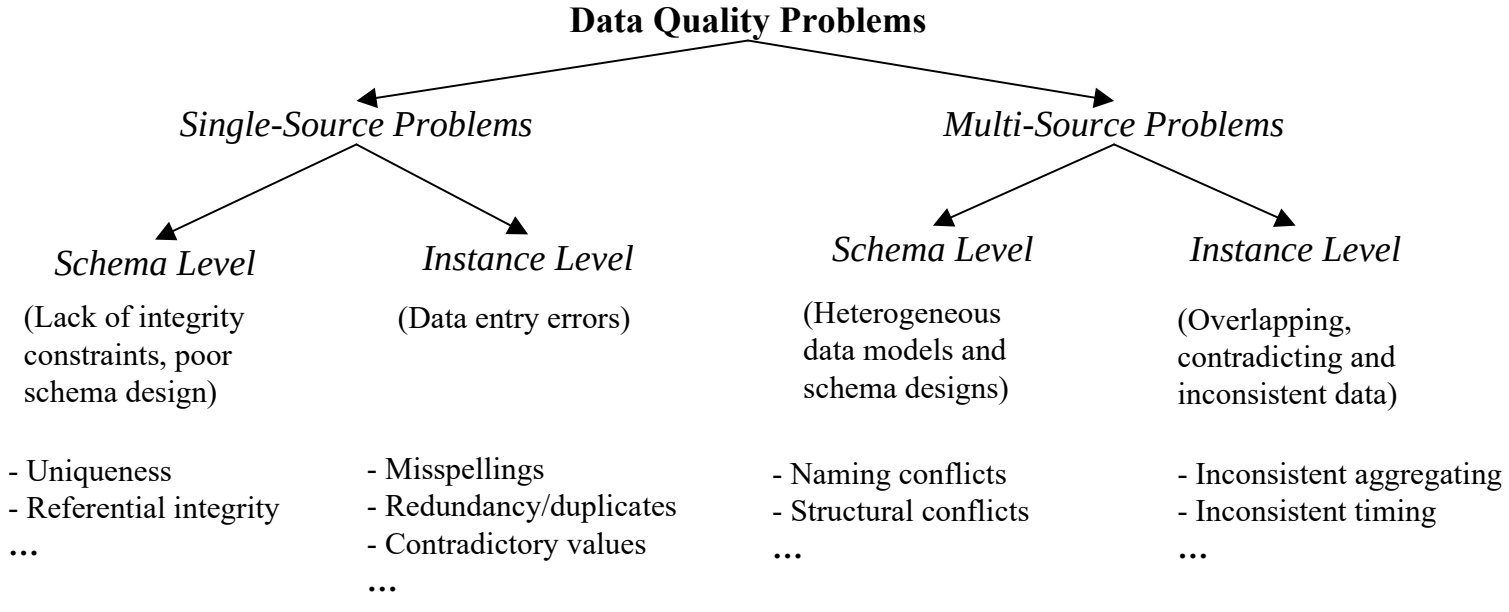
- Datenanalyse / Profiling
  - Entdeckung von Datenfehlern und -inkonsistenzen
  - manuell bzw. Einsatz von Analyse-Tools
- Definition von Mapping-Regeln und Transformations-Workflows
  - Datentransformationen auf Schemaebene
  - Cleaning-Schritte zur Behandlung von Instanzdaten
  - deklarative Spezifikation erlaubt automatische Generierung von ausführbaren Skripts
- Test / Verifizierung der Transformations-Workflows
  - Korrektheit und Effektivität auf Kopien/Ausschnitt der Daten
- Transformation
  - regelmäßige Ausführung der geprüften Transformationsschritte
- ggf. Rückfluss korrigierter Daten in operative Quellsysteme

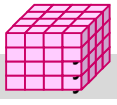
* E. Rahm, H. H. Do: *Data Cleaning: Problems and Current Approaches.* IEEE Techn. Bulletin on Data Engineering, Dec. 2000



Probleme bezüglich Datenqualität

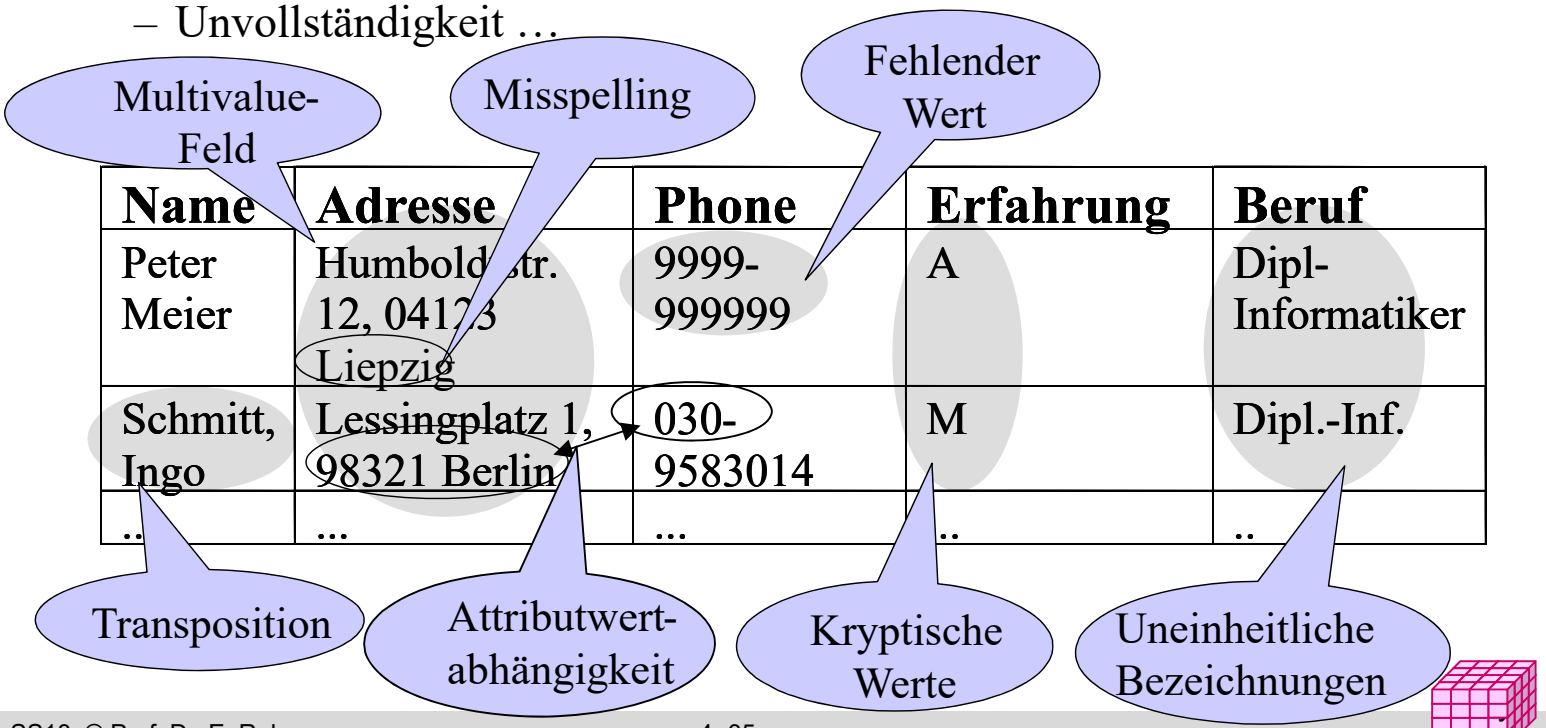
- Probleme auf Schema- und auf Instanzebene
- Probleme bezüglich einer oder mehrerer Datenquellen (Single-Source vs. Multi-Source)





Single-Source Probleme

- Ursachen:
  - Fehlen von Schemata (z.B. bei Dateien) und von Integritäts-Constraints
  - Eingabefehler
  - unterschiedliche Änderungsstände
  - Unvollständigkeit …



Multi-Source-Probleme

- überlappende, widersprüchliche bzw. inkonsistente Daten
  - aufgrund unabhängiger Erzeugung / Speicherung in verschiedenen Quellen
- Hauptproblem: Behandlung überlappender Daten
  - gängige Bezeichnungen: *Duplikate, Merge/Purge-Problem, Object Identity Problem, Record Linkage*
  - Beschreibung einer Instanz der realen Welt durch mehrere Datensätze unterschiedlicher Quellen
  - oft nur teilweise Redundanz (einzelne Attribute, nur in Teilmenge der Datenquellen) -> Fusion der Instanzen notwendig
- unterschiedliche Repräsentationen der Instanzdaten
  - versch. Wertebereiche (z.B. *Geschlecht = {1,2} vs. Gender = {m,w}*)
  - verschiedene Einheiten (z.B. *Verkauf in EUR vs. Verkauf in Tsd.EUR*)
  - verschiedene Genauigkeiten
- unterschiedliche Änderungsstände und Aggregationsstufen der Quelldaten

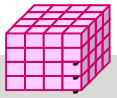
Multi-Source-Dateninkonsistenzen: Beispiel

Source1: Customer

CID	Name	Street	City	Sex
11	Kristen Smith	2 Hurley Pl	South Fork, MN 48503	0
24	Christian Smith	Hurley St 2	S Fork MN	1

Source2: Client

Cno	LastName	FirstName	Gender	Address	Phone/Fax
24	Smith	Christoph	M	23 Harley St, Chicago IL, 60633-2394	333-222-6542 / 333-222-6599
493	Smith	Kris L.	F	2 Hurley Place, South Fork MN, 48503-5998	444-555-6666



Beispiel (2)

CID	Name	Street	City	Sex
11	Kristen Smith	2 Hurley Pl	South Fork, MN 48503	0
24	Christian Smith	Hurley St 2	S Fork MN	1

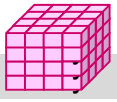
Source1: Customer

No	LName	FName	Gender	Street	City	State	ZIP	Phone	Fax	CID	Cno
1	Smith	Kristen L.	F	2 Hurley Place	South Fork	MN	48503-5998	444-555-6666		11	493
2	Smith	Christian	M	2 Hurley Place	South Fork	MN	48503-5998			24	
3	Smith	Christoph	M	23 Harley Street	Chicago	IL	60633-2394	333-222-6542	333-222-6599		24

Customers *(Integrierte und bereinigte Daten)*

Cno	LastName	FirstName	Gender	Address	Phone/Fax
24	Smith	Christoph	M	23 Harley St, Chicago IL, 60633-2394	333-222-6542 / 333-222-6599
493	Smith	Kris L.	F	2 Hurley Place, South Fork MN, 48503-5998	444-555-6666

Source2: Client



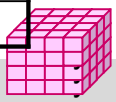
Datenanalyse / Profiling

- Entdeckung von Fehlern / Verifikation korrekter Werte
- Ableitung von (wirklichen) Metadaten
- Berechnung der Statistiken zu Attributen auf Basis ihrer Instanzen
  - Datentyp, Länge, Maximum und Minimum, Null-, Default-Werte, Kardinalität, ...
  - Ermitteln von Wertebereichen, Häufigkeiten und Mustern von Attributwerte
- Erkennung von Ausreißern, funktionalen Abhängigkeiten
  - SQL-Abfragen für Basis-Checks, zB

```
SELECT Stadt,count(*) as Anzahl From Student Group By Stadt order by 2
```

Attribute Values	#occurences
IBM	3000
I.B.M.	360
Intel Bus Mach	213
International Business Machine	36

Instanzwerte	Pattern	Identifizierte Datenkategorie
(978) 555-1212	(nnn) nnn-nnnn	Telefonnummer
036-55-1234	nnn-nn-nnnn	Social Security Number
abc@web.de	aaa@aaa.aa	Email-Addresse
12.03.2008	nn.nn.nnnn	Datum

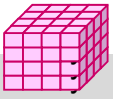


Behandlung von Single-Source-Problemen

- Definition und Einführung von Standardrepräsentationen
  - einheitliches Format für Datums-/Zeit-Angaben
  - einheitliche Groß/Kleinschreibungsform für Namen / String-Attribute
  - einheitliche Abkürzungen, Kodierungsschemas
- Bereitstellung von (Konversions-)Tabellen zur expliziten Werteabbildung

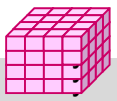
Legacy Value	New Value
IBM	IBM
I.B.M	IBM
Intel Bus Mach	IBM
...	...

- Extraktion von individuellen Werten aus Freiform-Attributen
  - Parsing und Attribut-Splitting, *z.B. Name -> Vorname / Nachname*
  - Reorganisierung der Wortreihenfolge
- Validierung / Korrektur mit Hintergrundwissen
  - Überprüfung/Spell checking mit Wörterbüchern, Datenbanken mit Adressen, Produktbezeichnungen, Akronymen/Abkürzungen, etc.
  - Nutzung bekannter Attributabhängigkeiten zur Korrektur von fehlenden / falschen Attributwerten



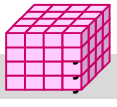
Behandlung von Multi-Source-Problemen

- Hauptproblem: Entdecken von Duplikaten bzw. korrespondierender Objekte (Objekt Matching)
- Durchführung auf aufbereiteten und gesäuberten Quellen
- Hauptschritte: Identifikation von "ähnlichen" Records (Matching) und Mischen (Merge) zu einem Record mit allen relevanten Attribute ohne Redundanz
- *Exact Matching*: Existenz eines Attributs oder eine Attributkombination zur eindeutigen Identifikation der einzelnen Records
  - Nutzung der Standard-Equijoin-Operationen zur Zusammenführung der zugehörigen Records
  - Sortierung der Records über die Schlüsselattribute und Vergleich der benachbarten Records zur Duplikatidentifikation
- *Fuzzy Object Matching*: keine gemeinsamen Schlüsselattribute (Normalfall)

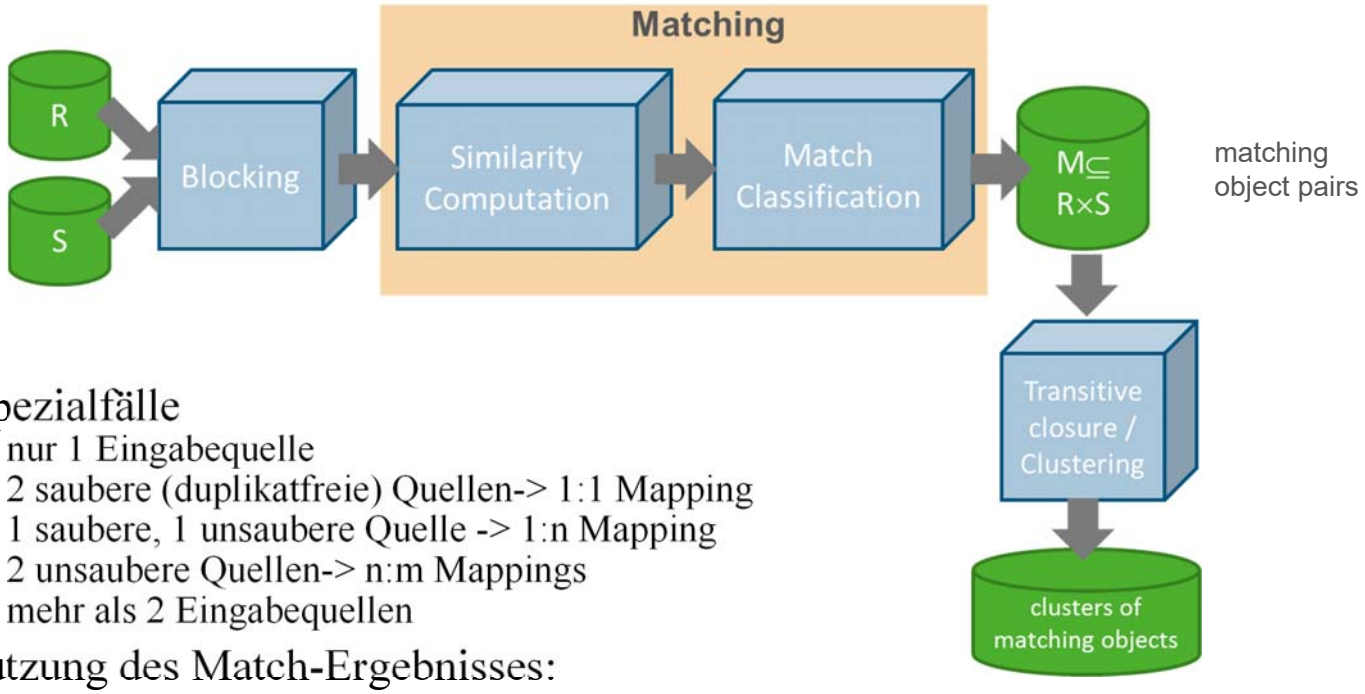


Fuzzy Object Matching

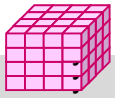
- erfordert meist kombinierte Nutzung mehrerer Ähnlichkeitsfunktionen und Erreichen einer Minimalähnlichkeit
  - Attributähnlichkeiten z.B auf Basis von Stringähnlichkeiten (Edot-Distance, Q-Gramme, TF/IDF, ...)
  - ggf. Berücksichtigung von Kontextinformationen (z.B. Gatte bei Personen, Koautoren bei Autoren, etc.)
  - Nutzung manueller Match-Regeln mit Gewichtung einzelner Ähnlichkeiten oder lernbasierte Match-Klassifikation (auf Basis von Trainingsdaten)
  - Beispiel: Personen-Matching auf Basis von Ähnlichkeiten für Name, Geburtsdatum und Adresse
- aktives Forschungsproblem zur Datenintegration



Allgemeiner Match-Workflow



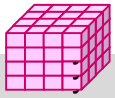
- Spezialfälle
  - nur 1 Eingabequelle
  - 2 saubere (duplikatfreie) Quellen-> 1:1 Mapping
  - 1 saubere, 1 unsaubere Quelle -> 1:n Mapping
  - 2 unsaubere Quellen-> n:m Mappings
  - mehr als 2 Eingabequellen
- Nutzung des Match-Ergebnisses:
  - Deduplizierung: Entfernung von Dubletten
  - Fusion sich ergänzender Attribute/Informationen



Fuzzy Object Matching (2)

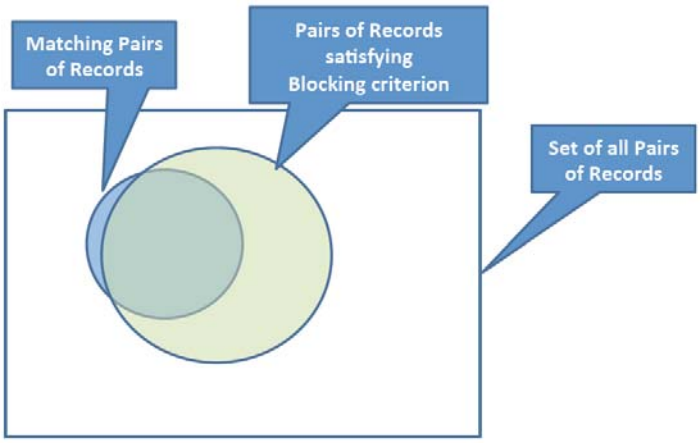
- Performance-Probleme für große Datenmengen
  - Auswertung des kartesischen Produkts von Objektpaaren (quadratische Komplexität) skaliert nicht
    - Bsp.: 1 Million Sätze, 1 Mikrosekunde pro Match
  - vorhergehendes Ausfiltern sehr unähnlicher Objektpaare zur Einschränkung zu prüfender Match-Kandidaten durch *Blocking* bzw. *Filter-Techniken*
  - *Paralleles Matching*, z.B. über Hadoop/MapReduce …
- Filter-Techniken: Nutzung von Eigenschaften bestimmter Ähnlichkeitsfuntionen und Schranken
  - optimiert für sogenannte *Similarity Joins* mit minimaler Ähnlichkeitsschranke t:

    $Sim\ (x,y) >= t$
  - Beispiele für q-Gram-Ähnlichkeit:
    - Strings ohne gemeinsame q-Gramme können nicht ähnlich sein
    - Strings mit stark unterschiedlicher Anzahl von q-Grammen können nicht ähnlich sein (Längenfilter)



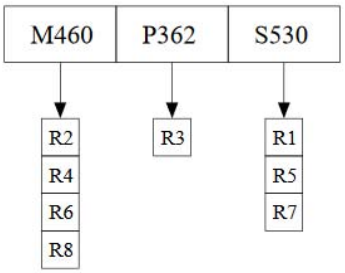
Blocking-Ansätze

- Nutzung bestimmter Attributwerte zur Partitionierung der Daten und Reduzierung des Suchraums
- Standard-Blocking
  - Partitionierung des Suchraums über Blocking Key Values (BKV)
  - BKV-Berechnung über Funktion auf Attributwerten, z.B. Soundex(Nachname), Postleitzahl, Prefix(Produkthersteller, 4), …
  - ggf. mehrere Durchgänge mit unterschiedlichen Keys (multi-pass-blocking)



Beispiel:

Identifiers	Surnames	BKVs (Soundex encoding)
R1	Smith	S530
R2	Miller	M460
R3	Peters	P362
R4	Myler	M460
R5	Smyth	S530
R6	Millar	M460
R7	Smyth	S530
R8	Miller	M460



P. Christen: A survey of indexing techniques for scalable record linkage and deduplication. IEEE TKDE, 2012

Blocking (2)

- Sorted Neighborhood
  - meist für 1 Eingabequelle
  - Sortierung gemäß *Sort Key*
  - Matching nur auf benachbarten Sätzen (Fenster fester Länge *w*)
  - lineare Komplexität

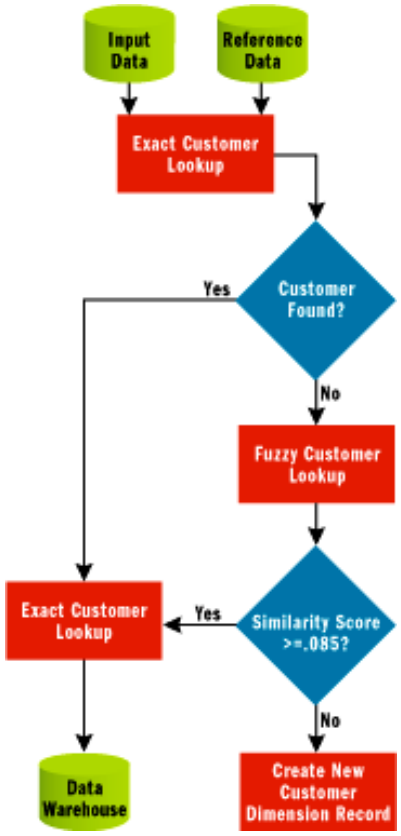
Window positions	BKVs (Surname)	Identifiers
1	Millar	R6
2	Miller	R2
3	Miller	R8
4	Myler	R4
5	Peters	R3
6	Smith	R1
7	Smyth	R5
8	Smyth	R7

Window range	Candidate record pairs
1 – 3	(R6,R2), (R6,R8), (R2,R8)
2 – 4	(R2,R8), (R2,R4), (R8,R4)
3 – 5	(R8,R4), (R8,R3), (R4,R3)
4 – 6	(R4,R3), (R4,R1), (R3,R1)
5 – 7	(R3,R1), (R3,R5), (R1,R5)
6 – 8	(R1,R5), (R1,R7), (R5,R7)

P. Christen: A survey of indexing techniques for scalable record linkage and deduplication. IEEE TKDE, 2012

MS SQL-Server: Data Cleaning Operatoren

- seit 2005 Bestandteil von SQL-Server Integration Services (SSIS; vormals DTS)*
  - Definition komplexer ETL-Workflows
  - zahlreiche Operatoren
- Fuzzy Lookup
  - "Fuzzy Join" zwischen Eingaberelation und sauberen Sätzen einer Referenztabelle: *inkrementelles Matching*
  - Parameter: Schwellwerte bzgl. String-Ähnlichkeit (Edit Distance) sowie Gewichte zur Kombination von Ähnlichkeiten
- Fuzzy Grouping
  - Gruppierung ähnlicher Sätze (potentielle Duplikate) innerhalb einer Tabelle über String-Matching (Edit Distance)



* http://msdn.microsoft.com/en-us/library/ms345128.aspx

Match-Prototypen (U Leipzig)

- MOMA (**M**apping based **O**bject **Ma**tching, 2007)
  - Unterstützung komplexer Match-Workflows mit mehreren Matchern
  - Nutzung bereits vorliegender Match-Ergebnisse
- FEVER (2009)
  - Weiterentwicklung im WDI-Lab, u.a. für Produkt-Matching
  - Unterstützung lernbasierter Verfahren zur vereinfachten Konfigurierung
  - Kommerzialisierung durch Spinoff
- DEDOOP (2012)
  - paralleles Objekt-Matching auf Hadoop
  - Lastbalancierung zur Behandlung von Skew-Effekten
- FAMER (2017)
  - Matching für viele (>2) Datenquellen
  - Matches werden in Clustern verwaltet
  - paralleles Matching auf Basis von Apache Flink

Duplikate in Webdaten: Beispiel

Canon VIXIA HF S10 Camcorder - 1080p - 8.59 MP - 10 x optical zoom
Flash card, 32 GB, 1y warranty, F/1.8-3.0
The VIXIA HF S10 delivers brilliant video and photos through a Canon exclusive 8.59 megapixel CMOS image sensor and the latest version of Canon's advanced image processor, ...
★★★★½ 12 reviews - Add to Shopping List

$975 new
from 52 sellers
Compare prices

Canon (VIXIA) HF S10 iVIS Dual Flash Memory Camcorder
Canon HF S10 iVIS Dual Flash Memory CamcorderSPECIAL SALE PRICE: $899 Display both English/Japanese + we supplu all English manuals in English as PDF. ....
Add to Shopping List

$899.00 new
Made in Japan Online

Canon VIXIA HF S10
Dual Flash Memory High Definition Camcorder The Next Step Forward in HD Video Canon has a well-known and highly-regarded reputation for optical excellence, ...
Add to Shopping List

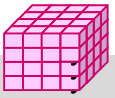
$999.00 new
Performance Audio
2 seller ratings

Canon VIXIA HF S100 Flash Memory Camcorder
***Canon Video HF S100 Instant Rebate Receive $200 with your purchase of a new Canon VIXIA HF S100 Flash Memory Camcorder. (Price above includes $200 ....
Add to Shopping List

$899.95 new
Arlingtoncamera.com
5 seller ratings

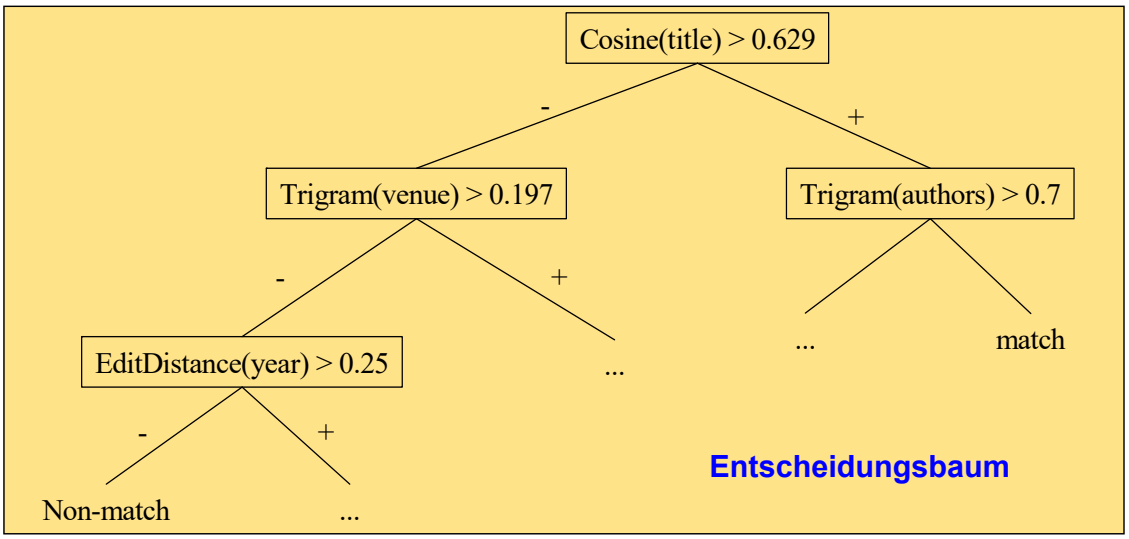
Canon Vixia Hf S10 Care & Cleaning
Care & Cleaning Digital Camera/Camcorder Deluxe Cleaning Kit with LCD Screen Guard Canon VIXIA HF S10 Camcorders Care & Cleaning.
Add to Shopping List

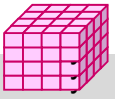
$2.99 new
shop.com
★★½☆☆ 38 seller ratings



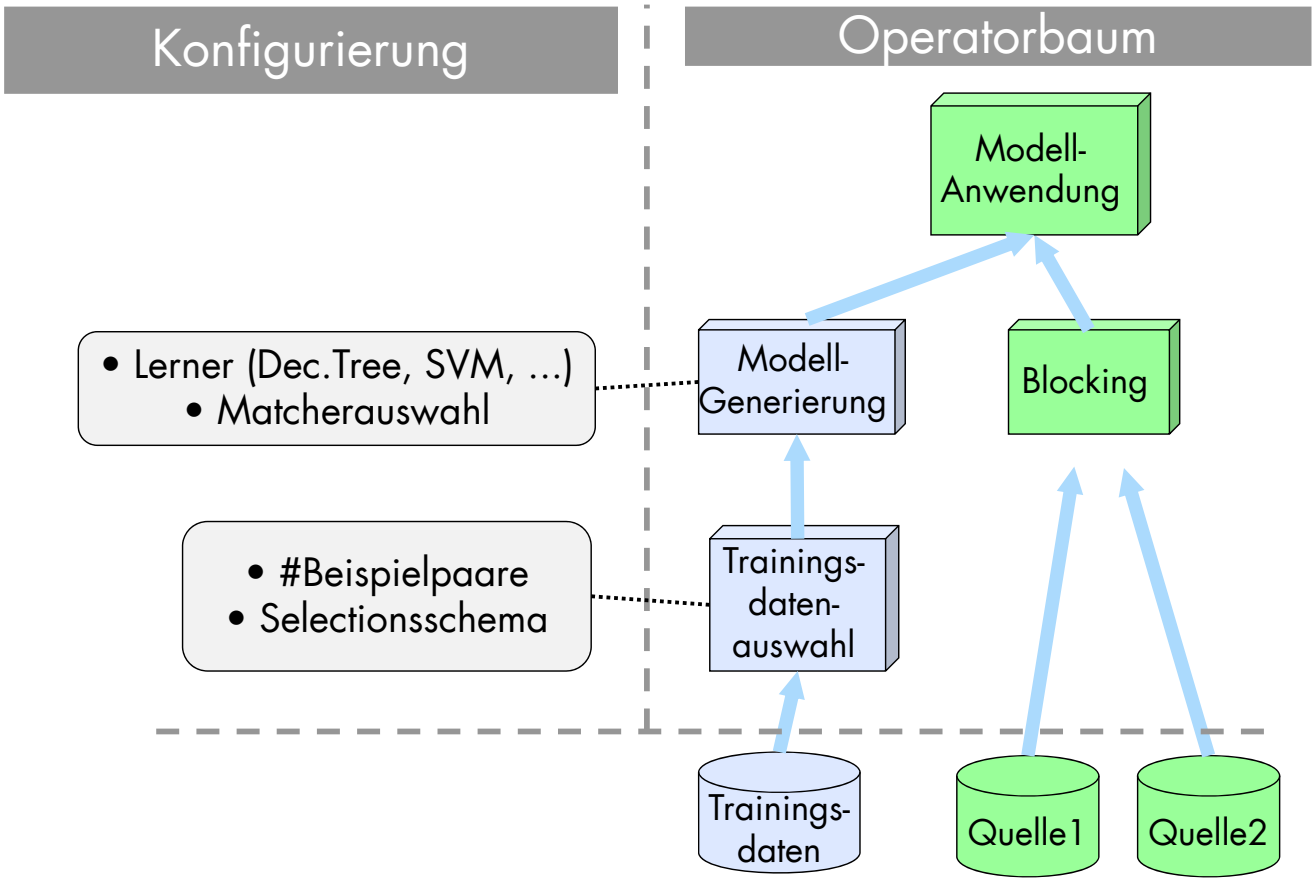
Trainingsbasiertes Objekt-Matching

- Finden effektiver Match-Einstellungen ist schwierig
  - Auswahl der Attribute, Matcher, Einstellungen
- Machine Learning verspricht Verbesserung
  - manuell spezifizierte Trainingsdatenmenge
  - Lernen von Match-Kriterien (z.B. mit Entscheidungsbaum)
  - erfordert gute Trainingsdaten mit vertretbarem manuellem Aufwand



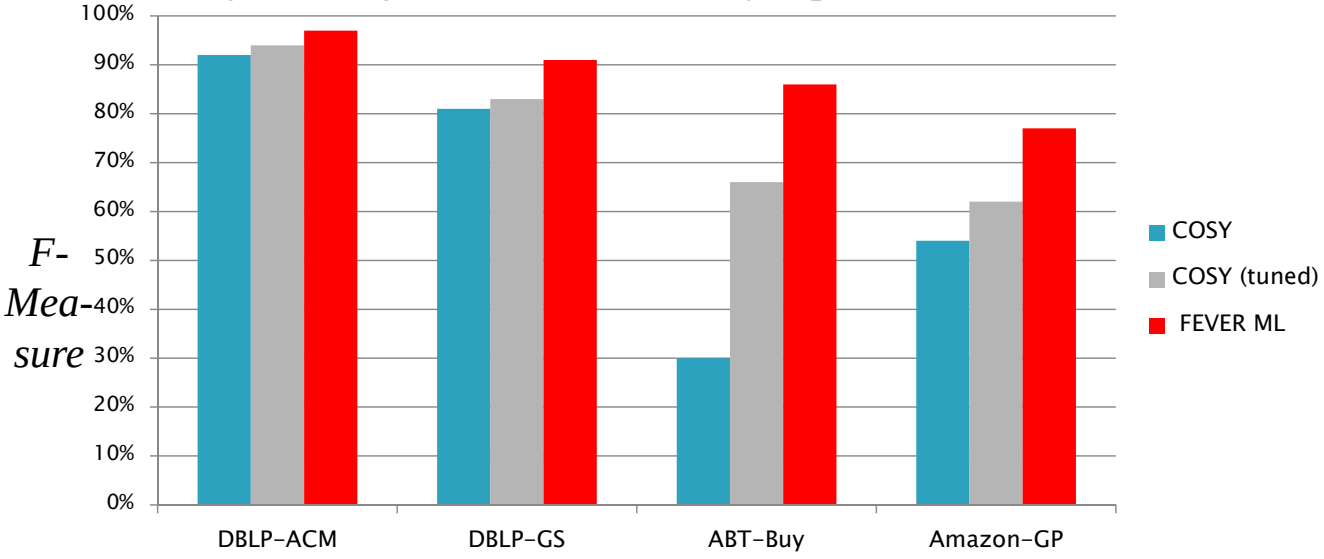


Lern-basierte Match-Strategien in FEVER



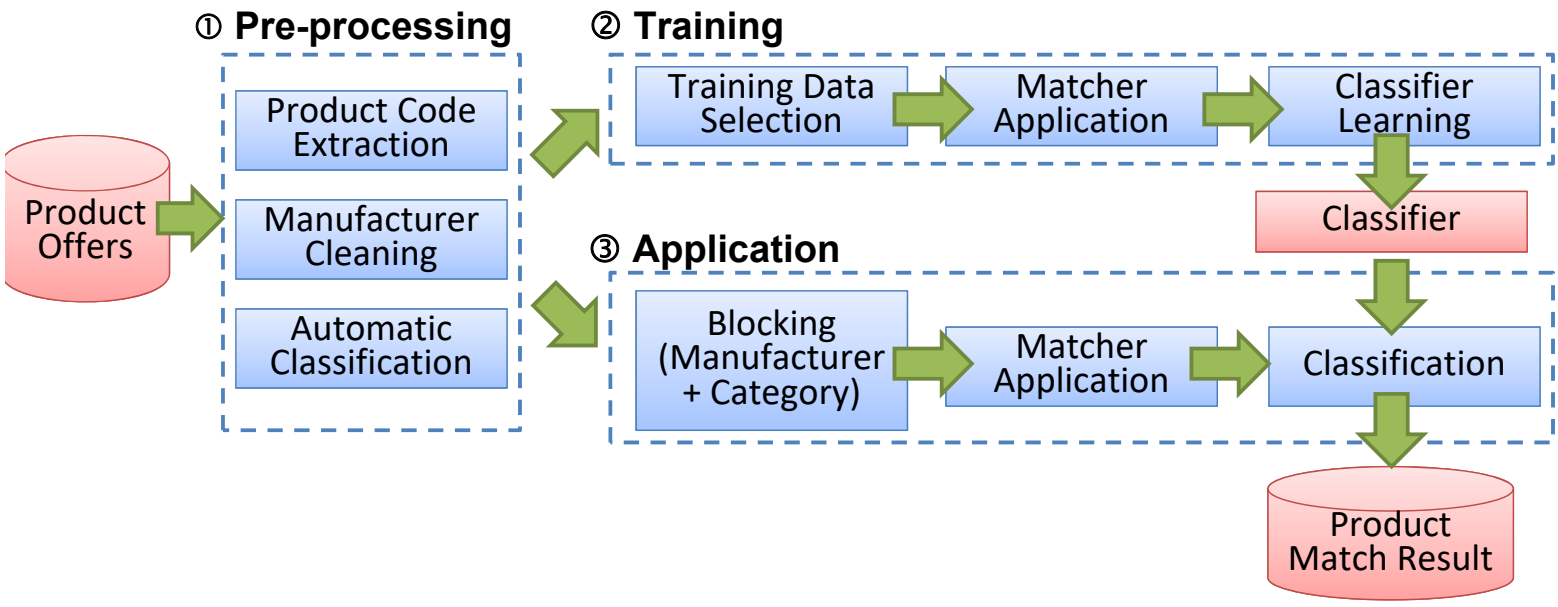
Evaluierungsergebnisse*

- 4 Matchaufgaben (bibliographisch, E-commerce)
- Matching auf 2 Attributen
- kommerz. System („COSY“) mit Default-Einstellungen und durch Fever optimierte Schwellwerte
- ML mit wenig Trainingsdaten (max. 500 Objektpaare)



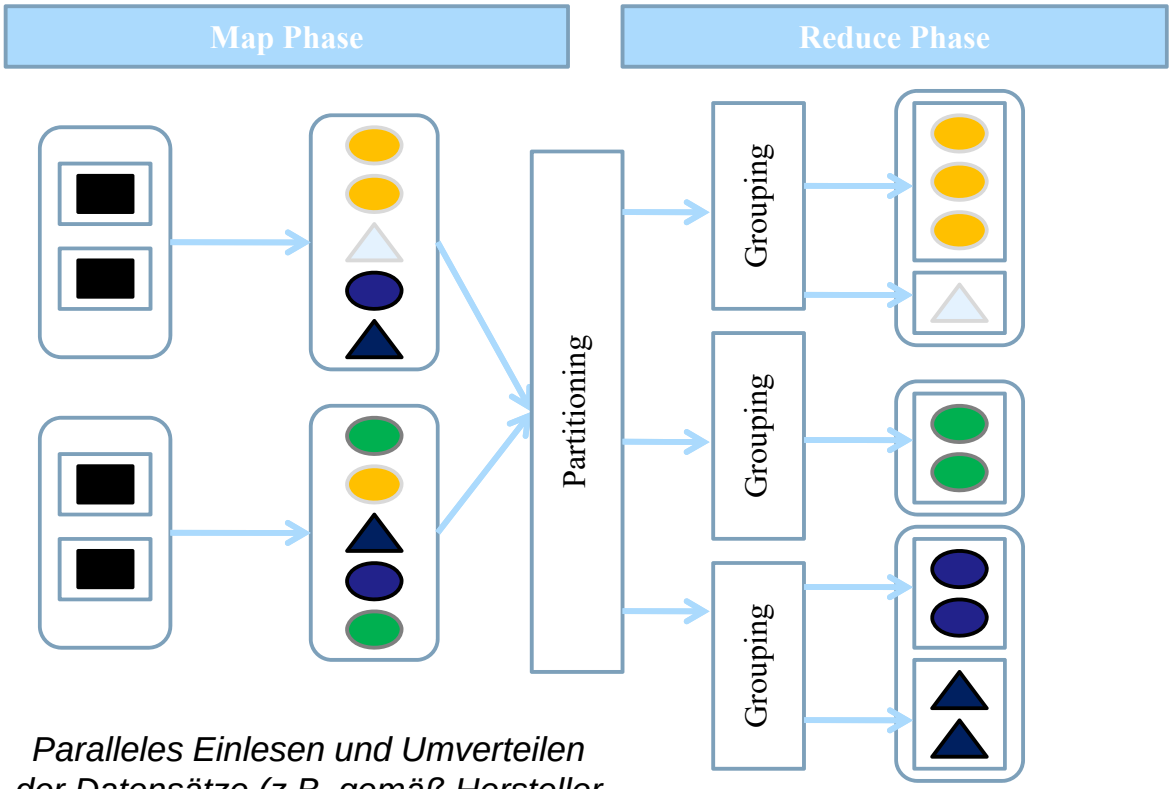
*Köpcke, H.; Thor, A.; Rahm, E.: ***Learning-based approaches for matching web data entities.*** IEEE Internet Computing, July 2010

Workflow zum Matching von Produktangeboten aus Web-Shops



Koepcke, Thor, Thomas, Rahm: *Tailoring entity resolution for matching product offers*. Proc. EDBT, 2012

Matching mit MapReduce



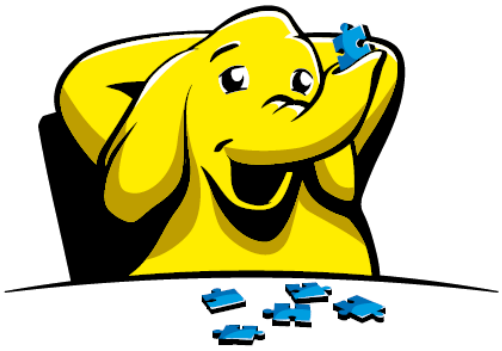
*Paralleles Einlesen und Umverteilen der Datensätze (z.B. gemäß Hersteller und Produkttyp) - **Blocking***

*Paralleler Abgleich pro Gruppe (**Matching**)*

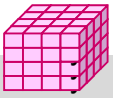
Dedoop: Efficient Deduplication with Hadoop

Parallele Ausführung von Datenintegrations/Match-Workflows mit Hadoop

- Browser-basiertes GUI
- mächtige Funktionsbibliothek mit
  - vielen Match-Techniken
  - lernbasierte Konfiguration
- automatische Generieren und Starten von Map/Reduce-Jobs auf unterschiedlichen Clustern
- automatische Lastbalancierung
- Monitoring der Ausführung

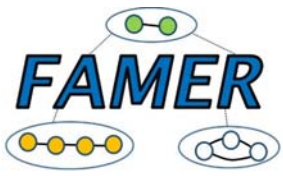


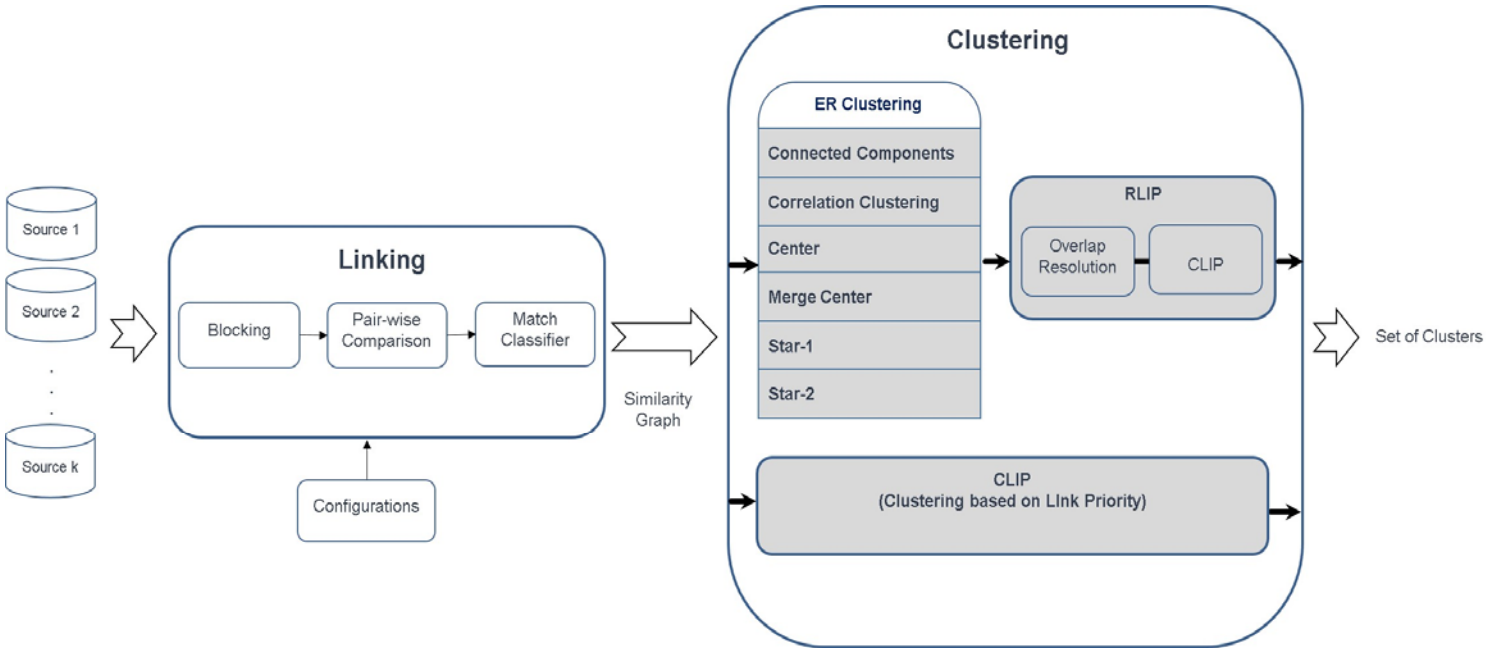
"This tool by far shows the most mature use of MapReduce for data deduplication"
www.hadoopsphere.com



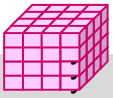
FAMER

- FAst Multi-source Entity Resolution system
  - Annahme: duplikatfreie Eingabequellen





https://dbs.uni-leipzig.de/research/projects/object_matching/famer

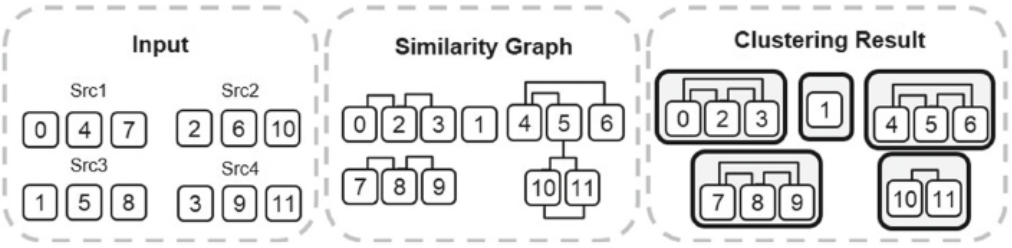


FAMER-Beispiel

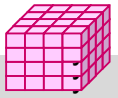
Eingabedaten aus 4 Quellen

Id	Name	Surname	Suburb	Post code	SourceId
0	ge0rge	Walker	winston salem	271o6	Src1
1	George	Alker	winstom salem	27106	Src2
2	George	Walker	Winstons	27106	Src3
3	Geoahge	Waker	Winston	271oo	Src4
4	Bernie	Davis	pink hill	28572	Src1
5	Bernie	Daviis	Pinkeba	2787z	Src2
6	Bernii	Davs	pink hill	28571	Src3
7	Bertha	Summercille	Charlotte	28282	Src1
8	Bertha	Summeahville	Charlotte	2822	Src2
9	Brtha	Summerville	Charlotte	28222	Src4
10	Bereni	dan'lel	Pinkeba	27840	Src3
11	Bereni	Dasniel	Pinkeba	2788o	Src4

FAMER-Anwendung

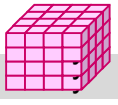


A. Saeedi, E. Peukert, E. Rahm. *Comparative evaluation of distributed clustering schemes for multi-source entity resolution*. Proc. ADBIS 2017.



Zusammenfassung (1)

- ETL als komplexer, aufwendiger Integrationsprozess
- Schema- und Datenintegration / Data Cleaning
  - zahlreiche Schema- und Datenkonflikte
  - begrenzte Automatisierbarkeit bezüglich Konflikterkennung und -behandlung
  - möglichst deskriptive Spezifikation aller Datentransformationen zur Behandlung von Schema- und Datenkonflikten
- Fokussierung auf Data Warehouse-spezifisches Zielschema erleichtert Schemaintegration
  - Top-Down-Schemaintegration
  - keine vollständige Integration aller Quell-Schemata erforderlich
- wichtiges Teilproblem: Schema-Matching
  - Nutzung und Kombination mehrerer Matcher, u.a. metadaten- und instanzbasierter Verfahren, linguistisches und strukturelles Matching
  - Reuse früherer Match-Ergebnisse
  - GUI
  - besondere Probleme für sehr große Schemas / Ontologien



Zusammenfassung (2)

- Unterscheidung quell-lokaler und -übergreifender Datenkonflikte
- Data Cleaning zunächst auf einzelnen Quellen
- zentrales Problem: Duplikat-Identifikation und –Behandlung (Objekt Matching)
  - hohe Effizienzanforderungen (-> Nutzung von Blockingverfahren und ggf. Parallelisierung)
  - kombinierte Nutzung mehrerer Match-Verfahren
  - besondere Herausforderungen für Webdaten
- trainingsbasierte Lernverfahren helfen bei der Konfigurierung
  - ermöglicht automatische Auswahl/Gewichtung von Matchern sowie Ähnlichkeitsschwellwerten
  - intelligente Auswahl der Trainingsdaten (Objektpaare mit Match-Entscheidung) erforderlich
  - viele Quellen: Match-Cluster als Basis zur physischen Fusionierung