Schema and Data Integration

Knowledge Objectives

1. Identify the problem of information integration
2. Enumerate the three solutions to information integration
3. Explain the three characteristics of a distributed database
4. Name five kinds of system heterogeneities
5. Name four kinds of semantic heterogeneities on instances
6. Name five kinds of semantic heterogeneities on classes
7. Name four kinds of structural semantic heterogeneities along generalization/specialization
8. Name four kinds of structural semantic heterogeneities along aggregation/decomposition

Understanding Objectives

1. Translate a global query into the corresponding local ones using GAV mappings
2. Given a similarity and merge functions, eliminate the duplicates in a table using the R-Swoosh algorithm

Application Objectives

1. Given two schemas of the same domain with structural discrepancies, decide whether they represent the same reality or not
2. Given a schema, propose an equivalent alternative showing any kind of semantic heterogeneity

PROBLEM DEFINITION

Flights example

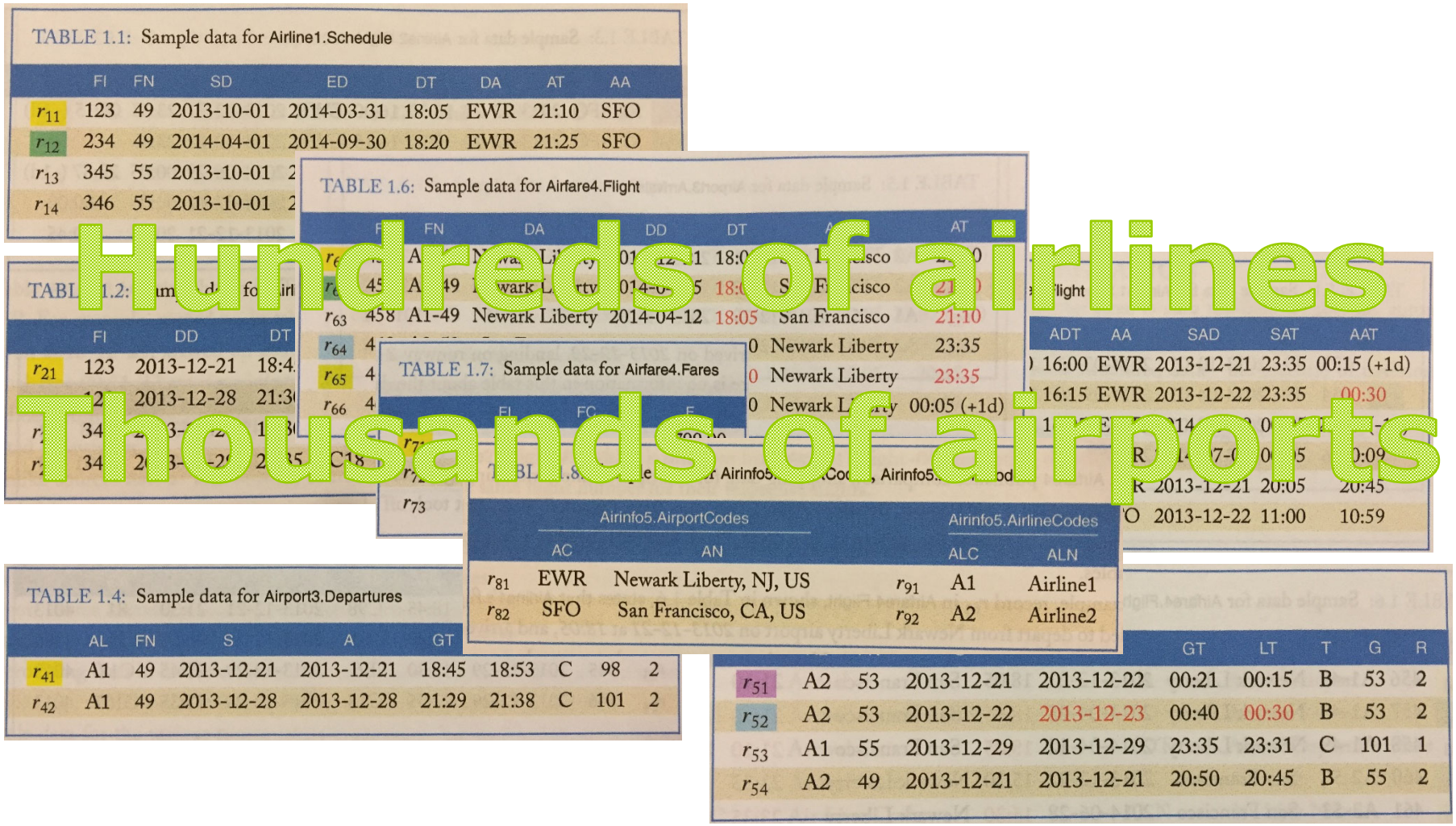


TABLE 1.1: Sample data for Airline1.Schedule

	FI	FN	SD	ED	DT	DA	AT	AA
r_{11}	123	49	2013-10-01	2014-03-31	18:05	EWR	21:10	SFO
r_{12}	234	49	2014-04-01	2014-09-30	18:20	EWR	21:25	SFO
r_{13}	345	55	2013-10-01	2				
r_{14}	346	55	2013-10-01	2				

TABLE 1.6: Sample data for Airfare4.Flight

	FN	DA	DD	DT	AT
r_{63}	458	A1-49	Newark Liberty	2014-04-12	18:05	San Francisco	21:10
r_{64}					0	Newark Liberty	23:35
r_{65}					0	Newark Liberty	23:35
r_{66}					0	Newark Liberty	00:05 (+1d)

TABLE 1.7: Sample data for Airfare4.Fares

	FI	FC	F
r_{73}			

	ADT	AA	SAD	SAT	AAT
	16:00	EWR	2013-12-21	23:35	00:15 (+1d)
	16:15	EWR	2013-12-22	23:35	00:30
		R		7-(	0:09
		R	2013-12-21	20:05	20:45
		O	2013-12-22	11:00	10:59

Airinfo5.AirportCodes

	AC	AN
r_{81}	EWR	Newark Liberty, NJ, US
r_{82}	SFO	San Francisco, CA, US

Airinfo5.AirlineCodes

	ALC	ALN
r_{91}	A1	Airline1
r_{92}	A2	Airline2

TABLE 1.4: Sample data for Airport3.Departures

	AL	FN	S	A	GT				
r_{41}	A1	49	2013-12-21	2013-12-21	18:45	18:53	C	98	2
r_{42}	A1	49	2013-12-28	2013-12-28	21:29	21:38	C	101	2

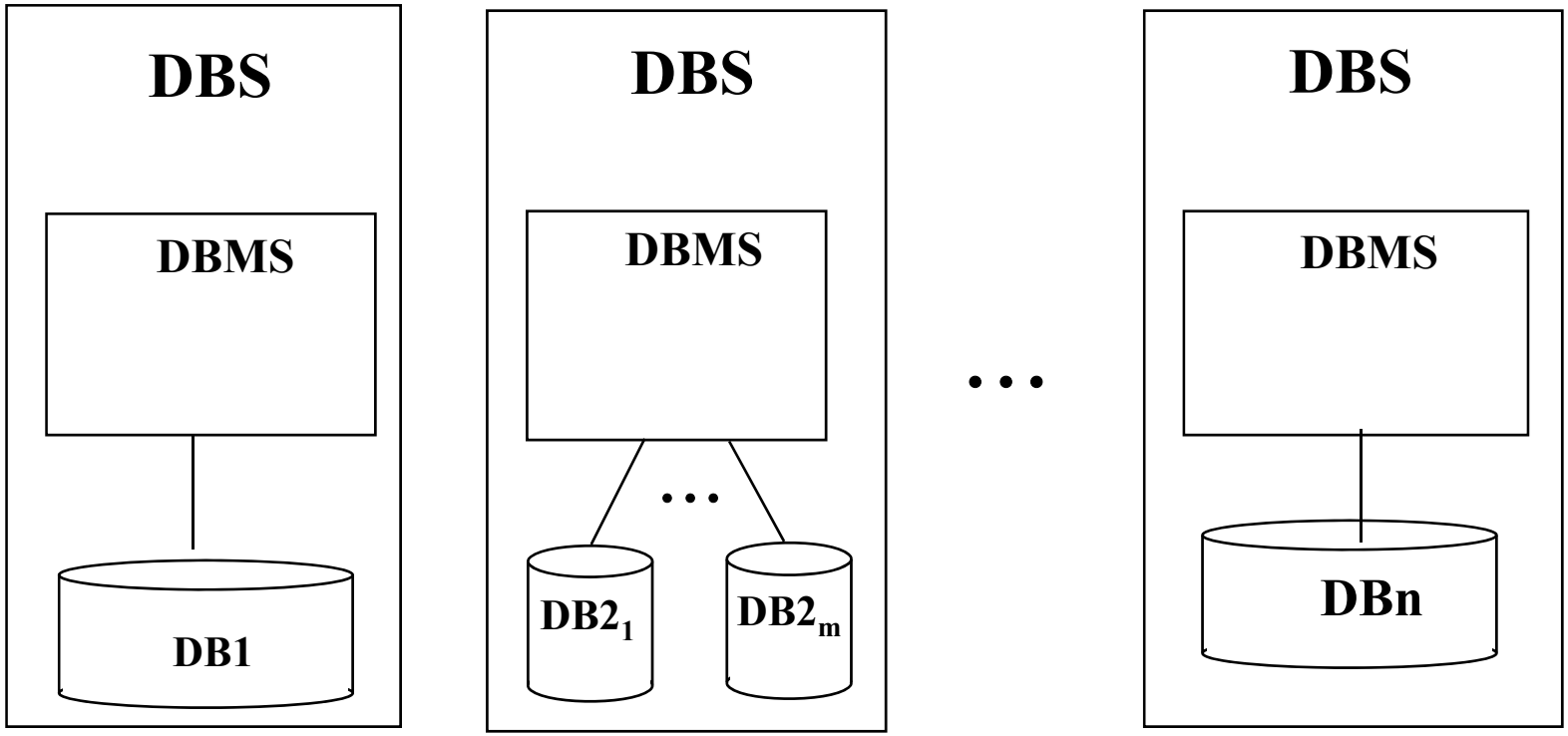
					GT	LT	T	G	R
r_{51}	A2	53	2013-12-21	2013-12-22	00:21	00:15	B	53	2
r_{52}	A2	53	2013-12-22	2013-12-23	00:40	00:30	B	53	2
r_{53}	A1	55	2013-12-29	2013-12-29	23:35	23:31	C	101	1
r_{54}	A2	49	2013-12-21	2013-12-21	20:50	20:45	B	55	2

X. L. Dong and D. Srivastava

The problem (I)

?

Answer a query that requires accessing several databases



The problem (II)

Being able to pose **one query**, and get **one answer**, so that in the preparation of the answer data coming from **several DBs** is processed.

It is not:

- BD connectivity (assumed)
- Electronic Data Interchange (EDI), Business to Business (B2B), eB-XML (p.ej. SOAP)
- Remote database access
- Multiclient/Multiserver architecture
- Distributed DBMS

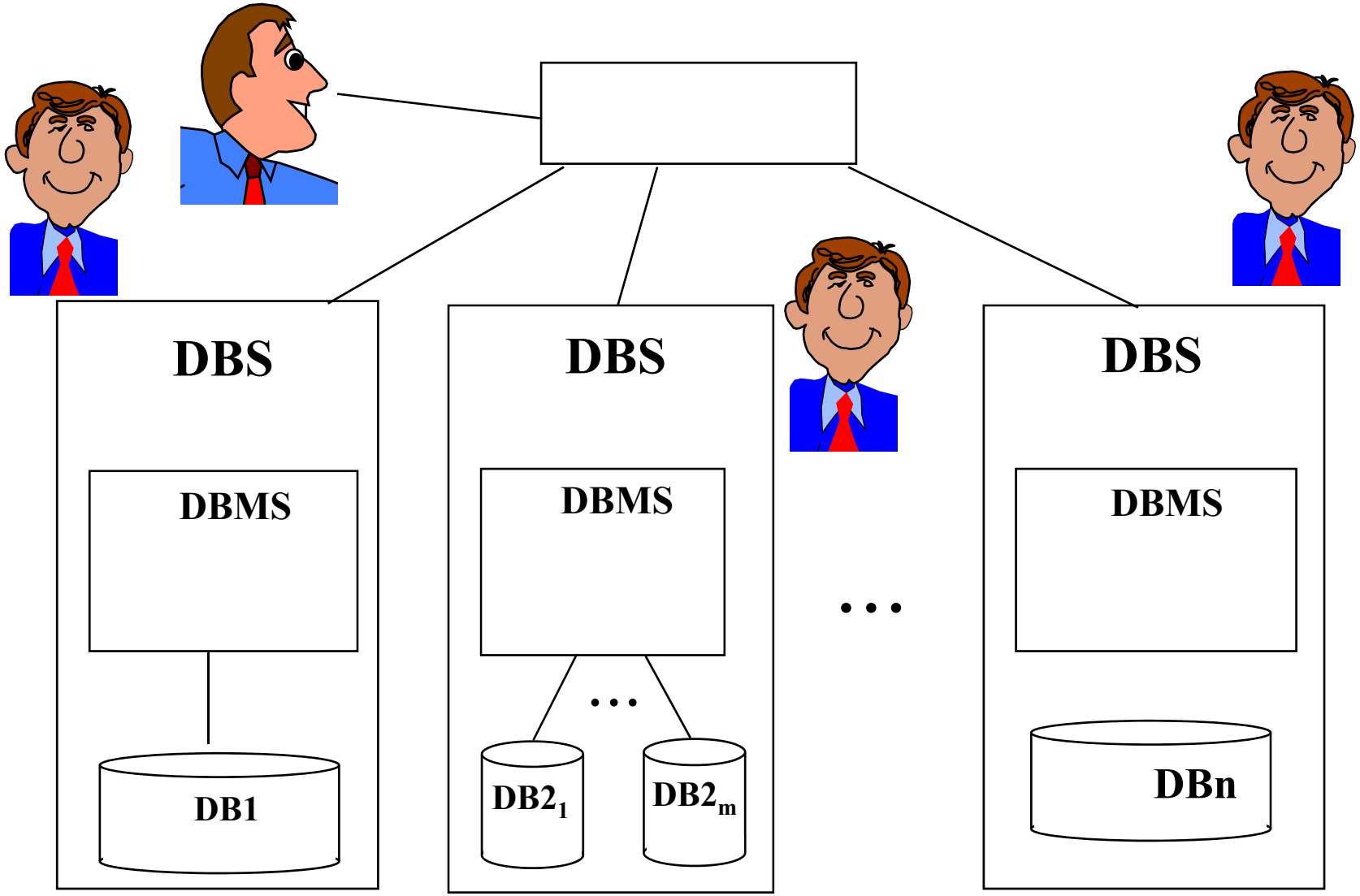
Solutions

a) Manually query the different databases separately
   - Know the available databases
     - Data
     - Data model
       - Query language
   - Decompose the query
   - Integrate the results

b) Create a new database containing all necessary data
   - Design it
   - Move data
   - Modify the applications to use the new repository
   - Test everything

c) Build a software layer on top of the databases that automatically splits the queries and integrates the answers
   - Add a new software layer that defines two access levels
   - Automatically process the queries

Users in the integrated system



Distributed Database

"A distributed database (DDB) is a collection of multiple, **logically interrelated** databases (known as nodes or sites) **distributed over a computer network**. A distributed database management system (DDBMS) is thus, the software system that permits the management of the distributed database and makes the **distribution transparent to the users**."

Tamer Özsu & P. Valduriez
Principles of DDB Systems
Springer, 2011

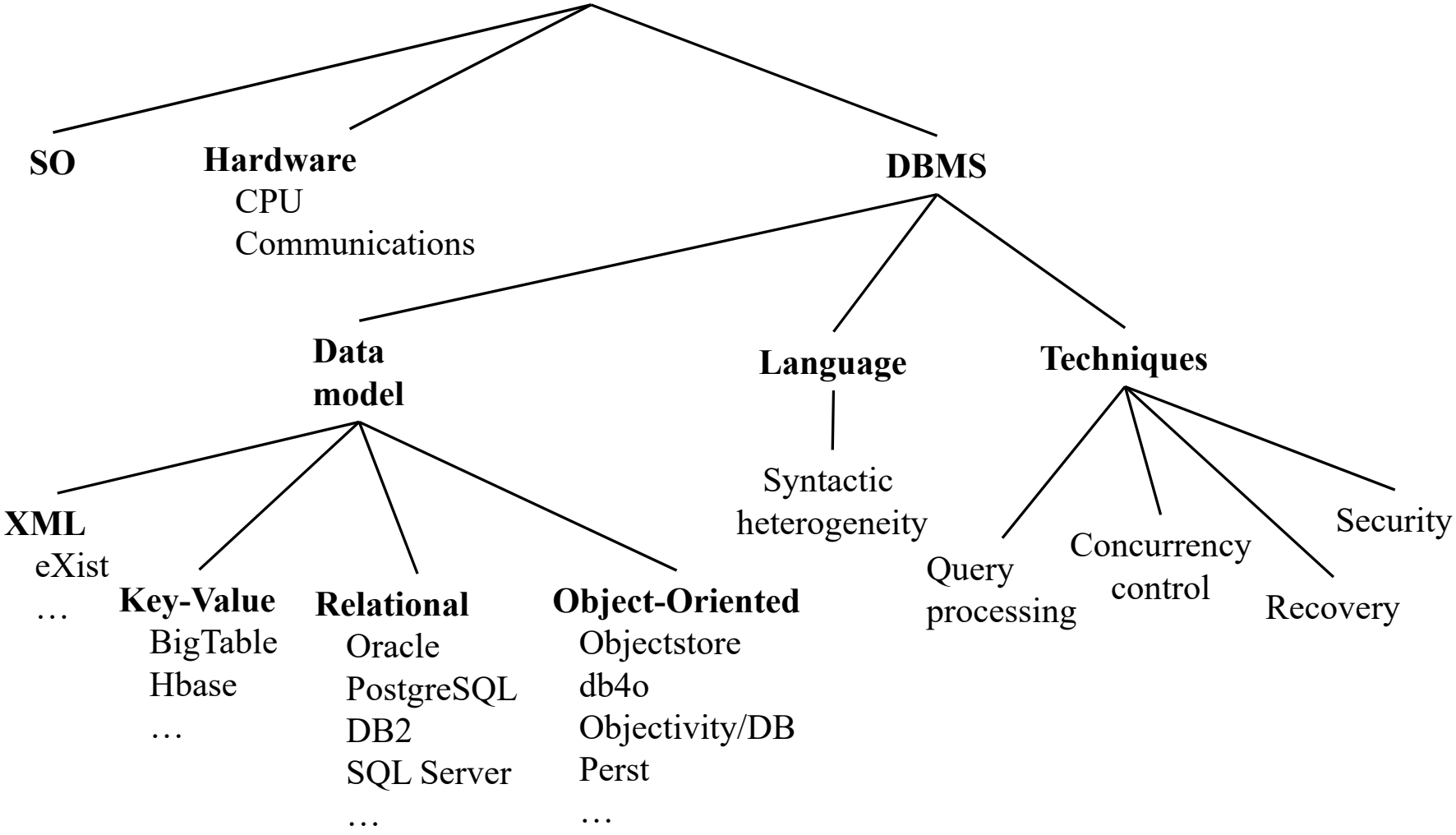
Classification of DDB

- Autonomy
  a) Design
  b) Execution
  c) Association
- Heterogeneity
  a) System
  b) Semantic

Polyglot persistence, Martin Fowler

HETEROGENEITY

System heterogeneities



Semantic heterogeneities: Instances

- Presence/Absence
- Number of values (multi/mono-valued)
- Existence of null values
- Value

Semantic heterogeneities: Classes

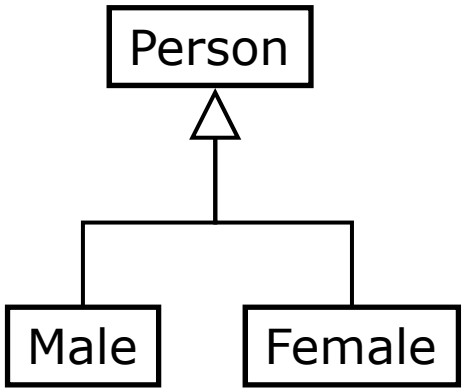
- Extension (e.g., coding colors)
- Name
- Attributes/Methods
  - Presence/Absence
  - Arity
  - Integrity constraints (e.g., mono/multi-valued)
- Domain
  - Keys
  - Data types
  - Dimension (e.g., volume vs weight)
  - Measuring units (e.g., liters vs gallons)
  - Scale (e.g., liters vs m^3)
- Constraints (checks and assertions)

Semantic heterogeneity: Structure

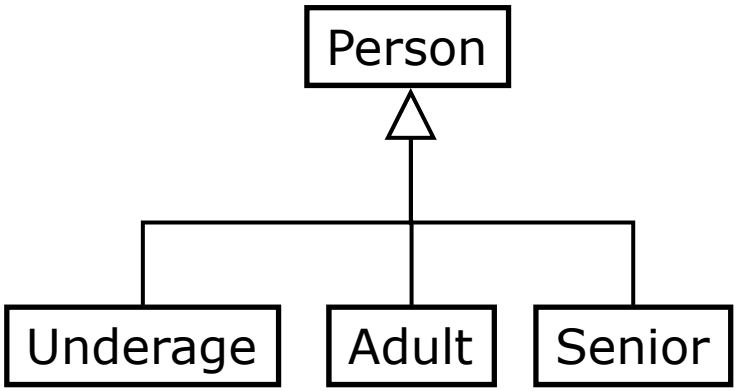
- Generalization/Specialization
  - Criterion (e.g., sex vs job)
  - Degree and characterization (e.g., different groups of age)
  - Kind (i.e., complete or not, disjoint or overlapping)
  - Integrity constraints (e.g., delete effect)
- Aggregation/Decomposition
  - Kind of aggregation (i.e., composition or not)
  - Participating classes
    - Specialization in the aggregated class (e.g., parent vs father)
    - Collection in the aggregated class (e.g., projects vs subprojects)
    - Composition in the aggregated class (e.g., address vs street+number+city)
  - Kind of partitioning collection (i.e., complete or not, disjoint or overlapping)
  - Component class of the collection (e.g., collection of counties vs collection of states)
- Schematic
  - Specialization vs Composition
  - Data vs Metadata

Example of specialization discrepancies

Option 1



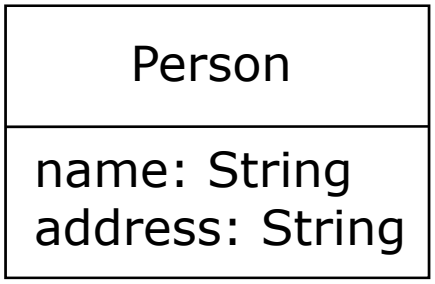
Option 2



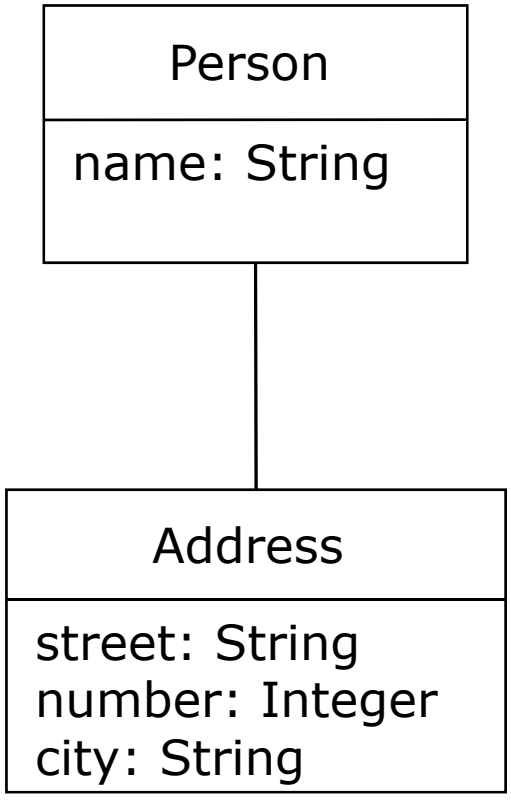
The class "Person" is the same, even if it has different subclasses

Example of aggregation discrepancies (I)

Option 1



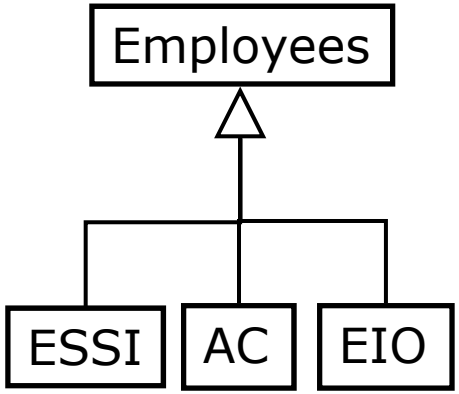
Option 2



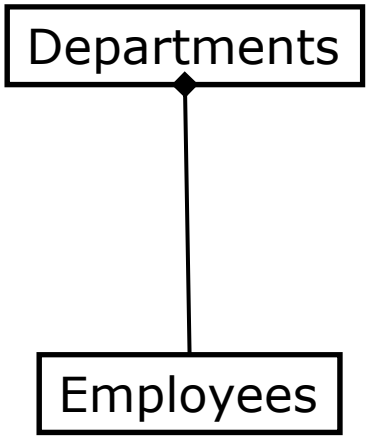
The class "Person" is the same, even if it has different attributes and associations

Example of schematic discrepancies (I)

Option 1



Option 2



These two alternative allow to represent basically the same reality, but what is considered metadata (i.e., subclasses of "Employees") at LHS is considered data (i.e., instances of "Departments") at RHS

Example of schematic discrepancies (II)

Option 1

Mothers
child: Person mother: Person

Fathers
child: Person father: Person

Option 2

Parenthood
child: Person father: Person mother: Person

Option 3

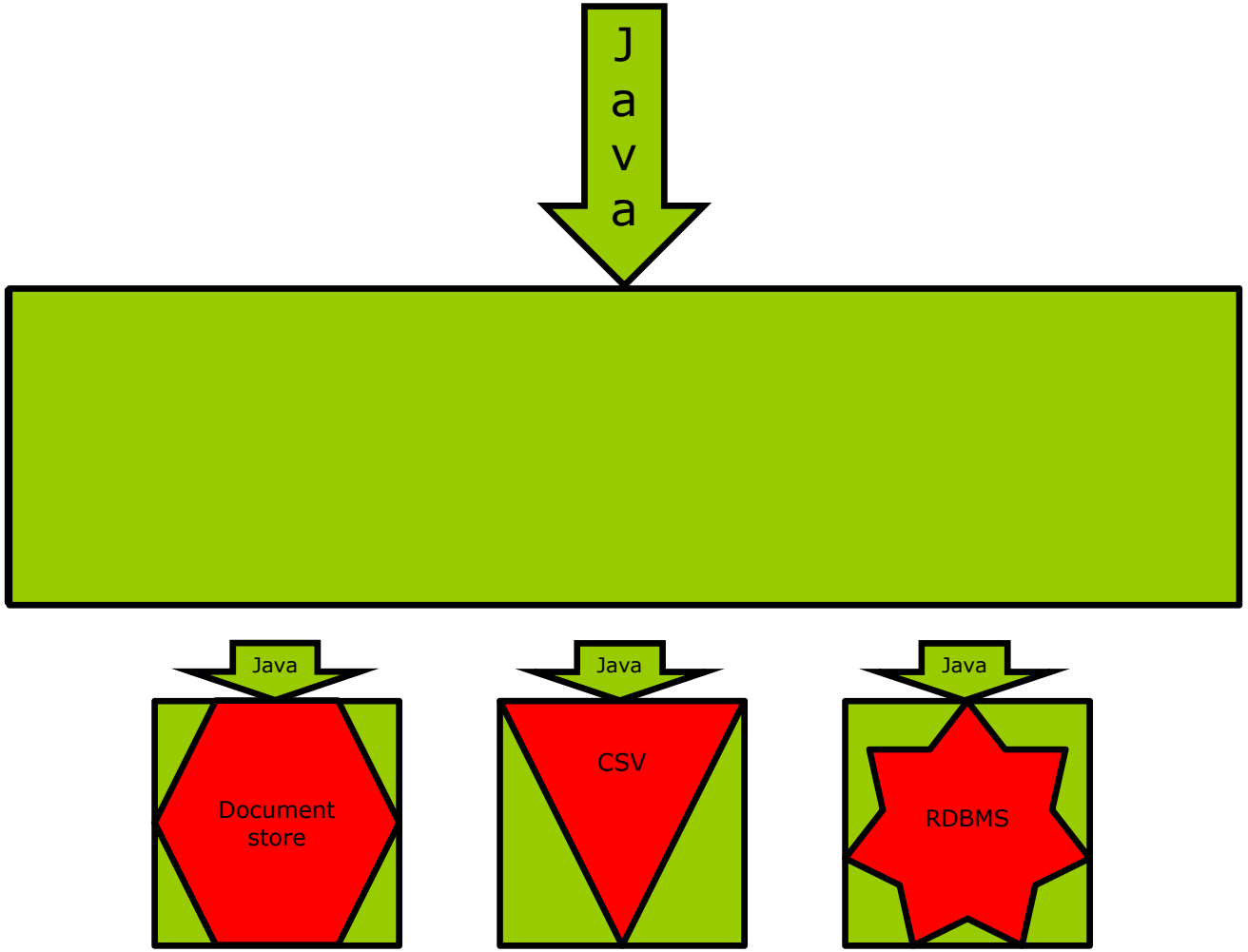
Parenthood
child: Person parent: Person kind: {Father, Mother}

These three alternative represent parenthood just with some subtle differences

Wrappers and Mediators

OVERCOMING HETEROGENEITY

Wrappers



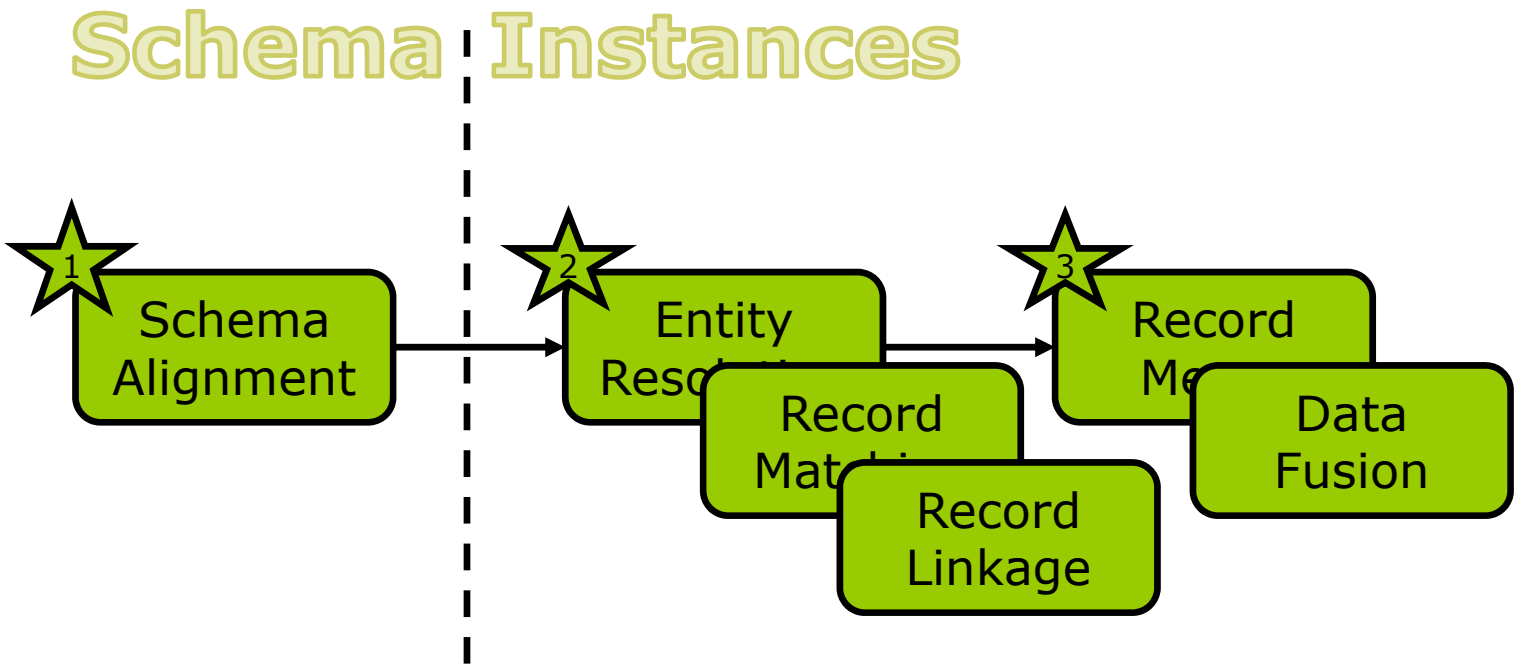
Wrapper example (CSV→Java)

```
public ArrayList<Employee> queryEmployees() {
   BufferedReader br = null;
   try {
      br = new BufferedReader(new FileReader("Employee.csv"));
      objects List<Employee> empList = new ArrayList<Employee>();
      String line = "";
      br.readLine();
      while ((line = br.readLine()) != null) {
         String[] employeeDetails = line.split(“,”);
         if(employeeDetails.length > 0 ) {
            Employee emp = new Employee(
               Integer.parseInt(employeeDetails[0]),
               employeeDetails[1],
               employeeDetails[2],
               Integer.parseInt(employeeDetails[3]));
            empList.add(emp);
            }
         }
      return empList;
      }
   catch(Exception ee) {
      throw new MyException(e);
      }
   }
```

Wrapper example (SQL→Java)

```
public ResultSet RetrieveR1 () throws MyException {
  try {
    Statement stmt = getConnection().createStatement();
    return stmt.executeQuery(“
      SELECT
        X.A AS title,
        1960+Y.B AS year,
        CASE WHEN Z.C = “Clean Eastboot”
        THEN “Clint Eastwood”
        ELSE Z.C END AS director
      FROM X, Y, Z
      WHERE X.FK=Y.ID AND Y.FK=Z.ID;
      ");
    } catch (Exception e) {
      throw new MyException(e);
    }
  }
```

Major steps to overcome semantic heterogeneities



Schema alignment and mapping

SCHEMA INTEGRATION

Alignment outcomes

a) Mediated/Integrated/Global schema
b) Attribute matching
   - Between sources and global
c) Schema mapping
   - Specify semantic relationships & transformations

Mappings

- Kinds
  - *Sound*

    $q_{Obtained} \subseteq q_{Desired}$
  - Complete

    $q_{Obtained} \supseteq q_{Desired}$
  - Exact

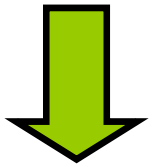
    $q_{Obtained} = q_{Desired}$
- Techniques
  - Global As View (GAV)
  - Local As View (LAV)
  - Global/Local As View (GLAV)
  - Peer To Peer (P2PDBMS)

Example of GAV (I)

- Global schema
  - Films (title, year, director)
  - EuropeanDirectors (director)
  - Reviews (title, review)
- Local schemas
  - European DB created in 1960
    - R1 (title, year, director)
  - World wide DB created in 1990
    - R2 (title, review, journalist)
- Mappings
  - Films (t, y, d):= R1[t, y, d]
  - EuropeanDirectors (d):= R1[d]
  - Reviews (t, r):= R2[t, r]

Example of GAV (II)

q(t, r) := Films [t, 1998] * Reviews [t, r]



Films (t, y, d) := R1[t, y, d]

q(t, r) := R1 [t, 1998] * Reviews [t, r]

Reviews (t, r) := R2[t, r]

q(t,r) := R1 [t, 1998] * R2 [t, r]

Characteristics of GAV

- The global schema is defined as a view over the data sources (bottom-up design)
- Query processing is absolutely equivalent to that in a centralized DBMS
- A mapping must be defined for each element in the global schema
- It can result too rigid, because any change in the data sources affects the global schema

Record matching and merging

DATA INTEGRATION

Integration

- Record matching (entity resolution)
  - Find an Object Identification Function
- Record merging
  - Schemas
  - Codification
  - Granularity
  - Units and scales
  - Data and metadata

Entity resolution

"Decide whether two tuples correspond to the same object in the real world."

- Problems to face:
  - Misspellings
  - Variant names
  - Misunderstanding of names
  - Evolution of values
  - Abbreviations
- Simplifications
  - Normalize the strings
  - Compare field by field

Similarity function (≈)

- Hamming distance
- Levenshtein distance
  - Edit distance
- Jaccard similarity
  - $\frac{|A \cap B|}{|A \cup B|}$

Merge function ($\wedge$)

- Solutions in case of different values
  - Generate null
  - Generate a multi-valued
  - Choose one on source trustworthiness
  - Use crowdsourcing

R-Swoosh algorithm

O:= ∅;
while (I<> ∅) do {
  pick up r∈I;
  if (∃s∈O so that s ≈ r) {
    I:=I-r; O:=O-s; I:=I∪{r ∧ s};
  } else {
    I:=I-r; O:=O∪{r};
  }
}

$s \approx r \Leftrightarrow Jaccard(s,r) \geq 0.3$
$r \wedge s = r \cup s$

I={ abc ac bc cd }

O={ }

CLOSING

Summary

- Distributed databases classification
- Heterogeneities
  - System
  - Semantic
- Schema alignment
  - Mappings
    - Global as View
- Entity resolution
  - R-swoosh algorithm
- Record merging

Bibliography

- O. A. Bukhres and A. K. Elmagarmid (Eds.). *Object-Oriented Multidatabase Systems*. Prentice-Hall, 1996
- H. Garcia-Molina et al. *Database Systems*. Prentice Hall, 2009
- T. Özsu and P. Valduriez. *Principles of Distributed Database Systems*. Springer, 2011
- X. L. Dong and D. Srivastava. Big Data Integration. Morgan&Claypool, 2015