Variety-aware analysis of document-oriented databases

Stefano Rizzi

DISI – University of Bologna, Italy

A joint work with **Enrico Gallinucci** and **Matteo Golfarelli**

Summary

Introduction

- NoSQL databases
- OLAP

Related work & contribution

(Explain: Schema profiling)

Analyze: Approximate OLAP

Conclusion

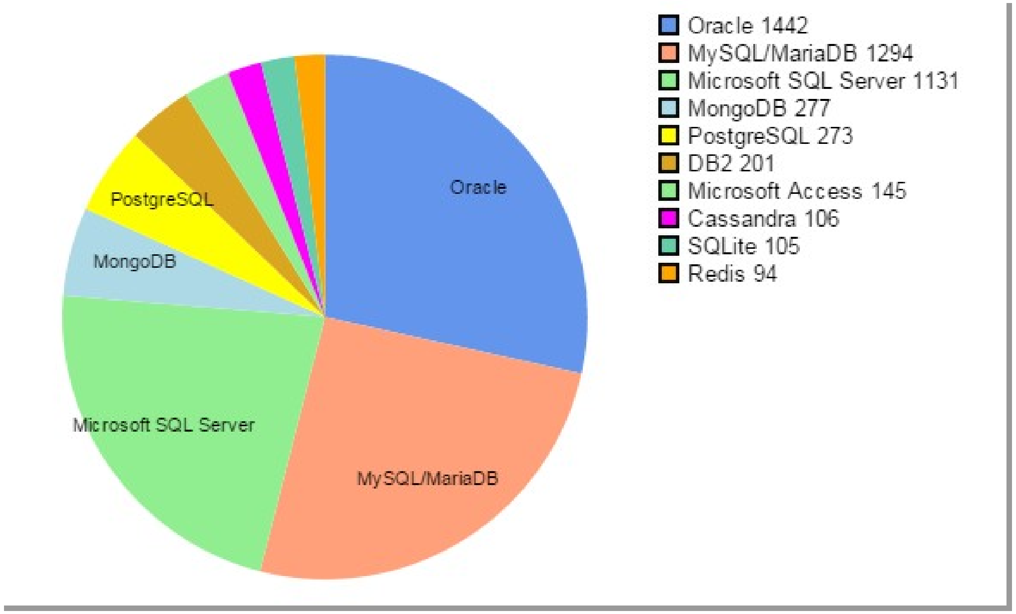
Introduction

In recent years, *NoSQL* databases have been progressively eroding the predominance of relational databases

DB-Engines Ranking, 2018

Market growth, 2019 (Gart

- RDBMSs: +15.2%
- NoSQL: +51.7%



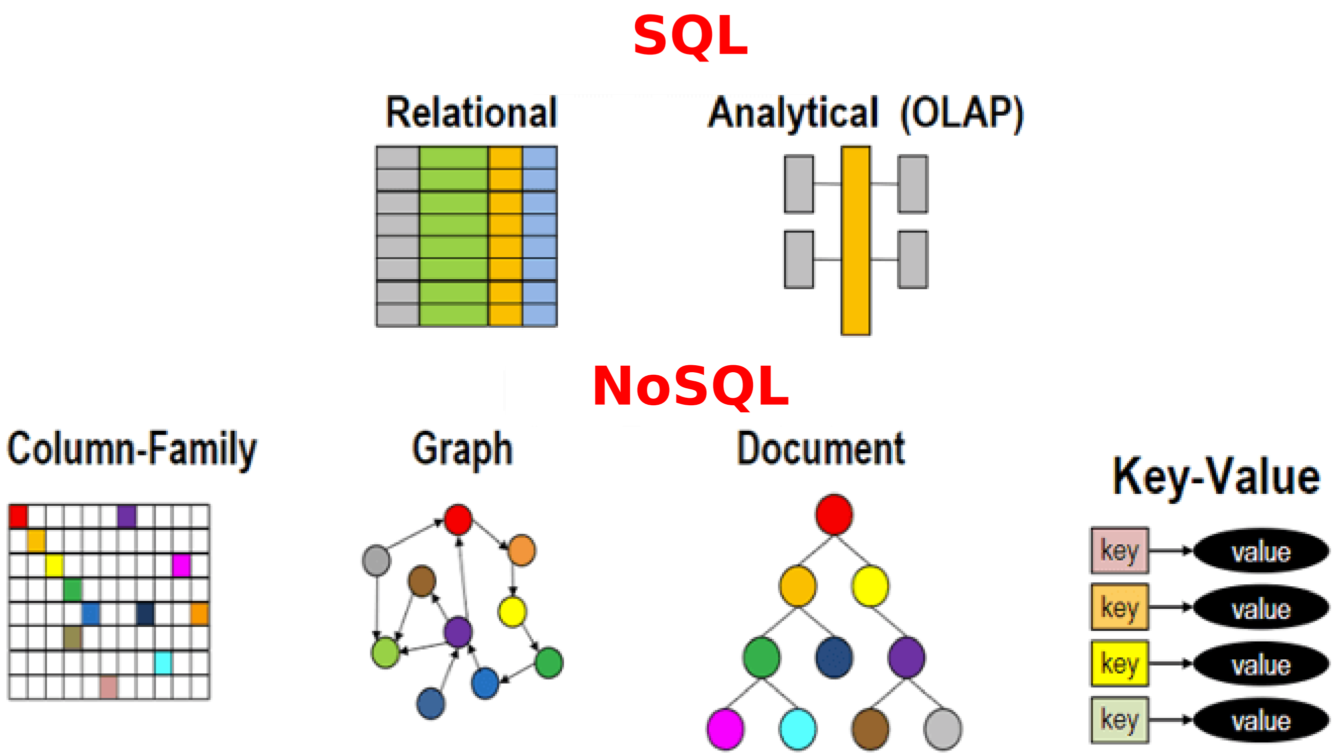
Introduction

A NoSQL database provides a mechanism for storage and retrieval of data that is modeled differently from the tabular relations used in relational databases

- key–value store
- columnar
- graph-based
- document-based

The particular suitability of a given NoSQL database depends on the problem it must solve

Introduction



Introduction

Why NoSQL?

- Better scaling
- No ACID transactions
- No need for a unique schema

Most new models adopt a schemaless representation for data

- Schema is a “soft” concept and the instances referring to the same concept can be stored using different *local* schemas
- Schemaless databases are preferred for storing heterogeneous data with variable schemas, such as those located in *data lakes*

Introduction

Typical schema variants that can be found within a NoSQL database consist in

- missing or additional attributes
- different names or types for an attribute
- different nesting

The absence of a unique schema gives flexibility to operational applications, but…

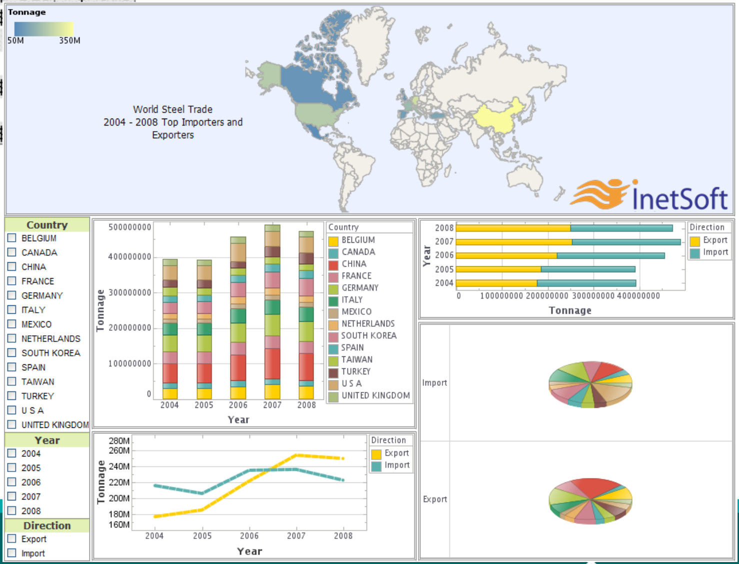
- … some extra effort is required to understand the rules that drove the use of alternative schemas
- …there is more complexity in analytical applications and OLAP, where queries often involve instances with different (possibly conflicting) schemas

Introduction

OLAP (On-Line Analytical Processing)

- Dynamic, multidimensional analyses that need to read a huge quantity of data to compute a set of numbers summing up the performance of a company

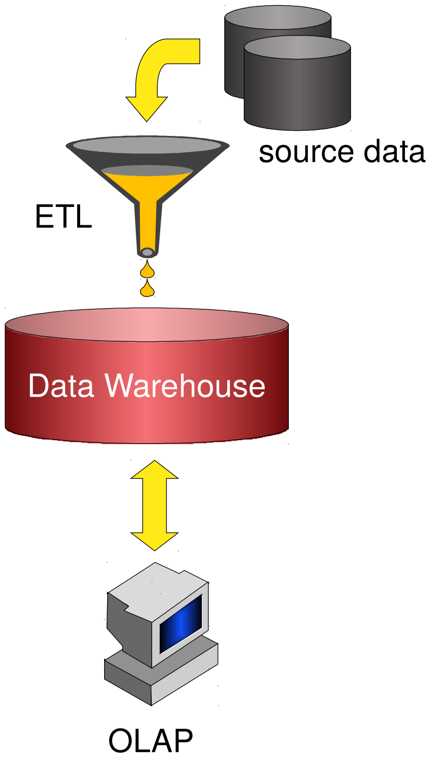
Metrics	Dollar Sales										
Customer Region / Quarter	North-East	Mid-Atlantic	South-East	Central	South	North-West	South-West	England	France	Germany	Canada
Q1 1997	$ 1.526	$ 1.249	$ 978	$ 1.885	$ 3.650	$ 4.855	$ 1.834	$ 2.552	$ 1.920	$ 643	$ 663
Q2 1997	$ 2.979	$ 1.415	$ 2.664	$ 1.582	$ 1.130	$ 1.906	$ 884	$ 1.393	$ 1.402	$ 516	$ 975
Q3 1997	$ 3.016	$ 1.772	$ 5.076	$ 2.050	$ 1.443	$ 2.311	$ 2.321	$ 608	$ 575	$	
Q4 1997	$ 2.711	$ 5.030	$ 2.025	$ 1.961	$ 3.601	$ 2.112	$ 3.976	$ 918	$ 531	$	
Q1 1998	$ 5.288	$ 3.399	$ 4.935	$ 9.174	$ 3.601	$ 9.484	$ 3.844	$ 2.720	$ 979	$	
Q2 1998	$ 1.773	$ 3.658	$ 1.862	$ 1.213	$ 1.115	$ 4.635	$ 352	$ 724	$ 2.110	$	
Q3 1998	$ 1.818	$ 5.402	$ 1.709	$ 2.485	$ 1.704	$ 3.632	$ 4.329	$ 679	$ 1.198	$	
Q4 1998	$ 2.051	$ 2.968	$ 4.664	$ 5.917	$ 1.775	$ 3.162	$ 3.485	$ 2.750	$ 1.005	$	



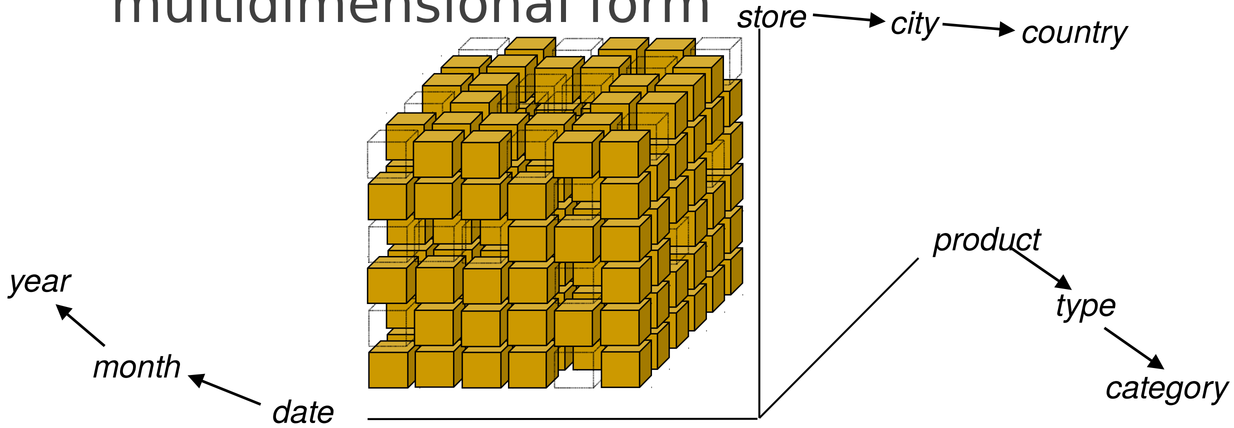
Introduction

Data Warehouse

- a repository of information that collects and integrates data coming from different, heterogeneous sources making them available for analyses aimed at planning and decision making



In a data warehouse, data are stored in multidimensional form



Related work

Schema discovery from XML/JSON documents

- dealing with heterogeneity, data quality, versioning, similarity, comprehensiveness...
- ...to produce unified schemas, schema matches, skeletons...
- ...to be used for querying, integration, validation

Schema matching for XML/JSON documents

- using clustering, machine learning...
- ...possible considering a context

OLAP analysis of document collections

- schema-on-write, schema-on-read...
- ...but no (or limited) management of variety

Our approach

Stop fighting against schema variety and welcome data heterogeneity as an inherent source of information wealth in schemaless sources

- focus on collections of documents in document-oriented databases

1. Schema profiling, to explain the schema variants within a collection by capturing the hidden rules explaining the use of these variants[1]
2. Approximate OLAP, to enable multidimensional querying of collections with variable schemas[2]

Related work & contribution

Schema profiling

Explain the schema variants within a collection by capturing the hidden rules explaining the use of these variants

Useful to...

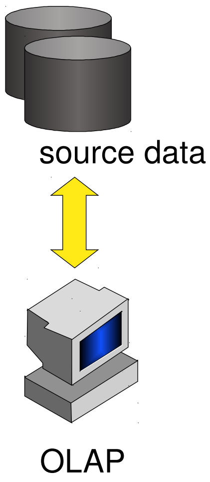
- ...decode the behavior of an undocumented application that manages a document-base
- ...carry out a data quality project on schemaless data
- ...enable a schema-on-read approach to query a document-oriented database
- ...design a data warehouse on top of a schemaless data source

Approximate OLAP

Enable multidimensional querying of collections with variable schemas

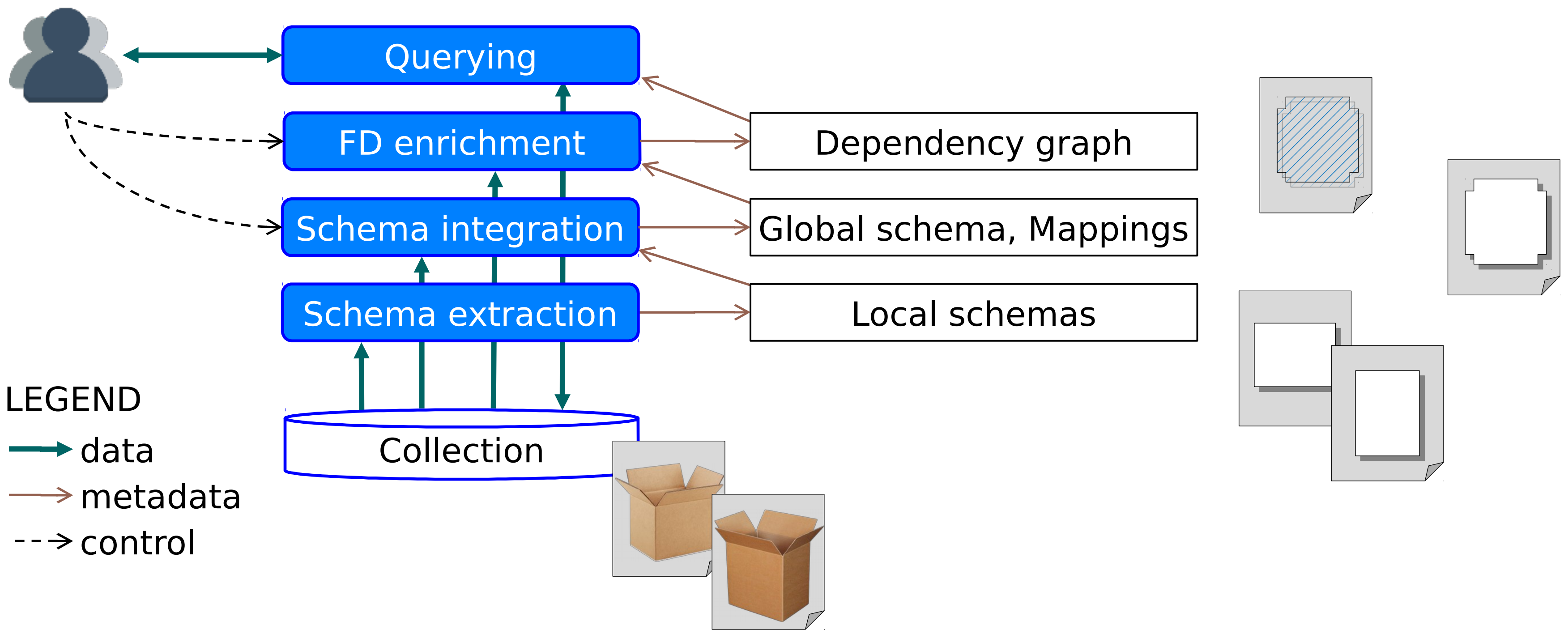
OLAP querying on a “soft” schema where each source attribute is present to some extent

- First variety-aware approach for approximate OLAP on document-oriented databases
- Querying is directly carried out on the data source (no cube materialization)
- Inclusive solution to integration
- Deal with both inter-schema and intra-schema variety
- Query reformulation on heterogeneous documents builds on a formal approach, which ensures its correctness and completeness



13

Overview



Approximate OLAP

Schema extraction

Goal: find the (local) schema of a document

```
[   {   "_id" : ObjectId("54a4332f44cfc02424f961d4"),
        "User" :
        {   "FullName" : "John Smith",
            "Age" : 42  },
        "StartedOn" : ISODate("2017-06-15T10:20:44.000Z"),
        "Facility" :
        {   "Name" : "PureGym Piccadilly",
            "Chain" : "PureGym"  },
        "SessionType" : "RunningProgram",
        "DurationMins": 90,
        "Exercises" :
        [   {   "Type" : "Leg press",
                "ExCalories" : 28,
                "Sets" :
                [   {   "Reps" : 14,
                        "Weight" : 60  },
                    . . .
                ] },
            {   "Type" : "Tapis roulant" },
            . . .
        ]
    } ,
    . . .
]
```

WS

_id
User.FullName
User.Age
StartedOn
Facility.Name
Facility.Chain
SessionType
DurationMins

Exercises

Exercises_id
Type
ExCalories

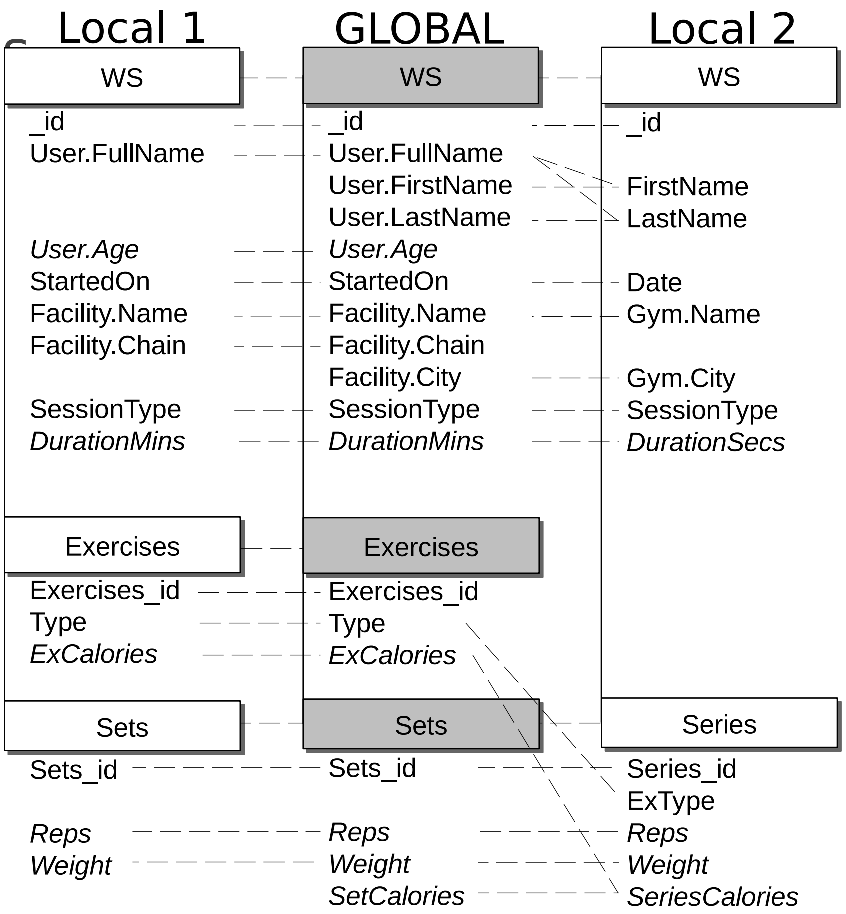
Sets

Sets_id
Reps
Weight

Schema integration

Goal: integrate the local schemas to obtain a s
schema

Integration through mappings



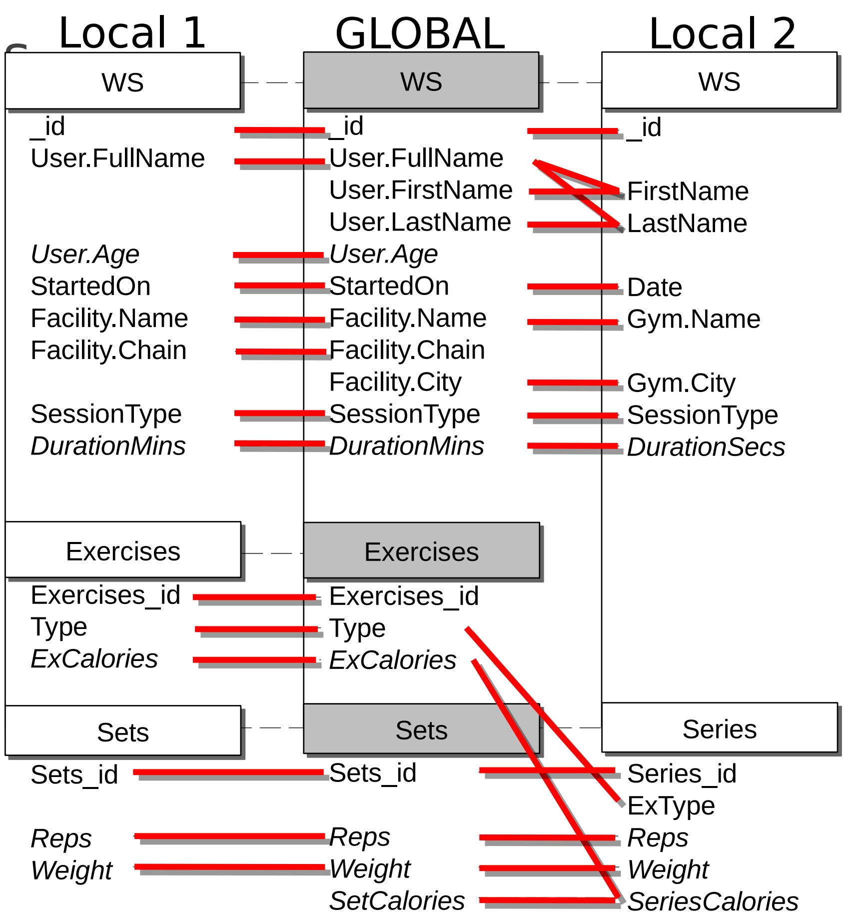
Approximate

Schema integration

Goal: integrate the local schemas to obtain a s
schema

Integration through mappings

- Primitive mappings
  - Only **exact** mappings
  - Transcoding functions required



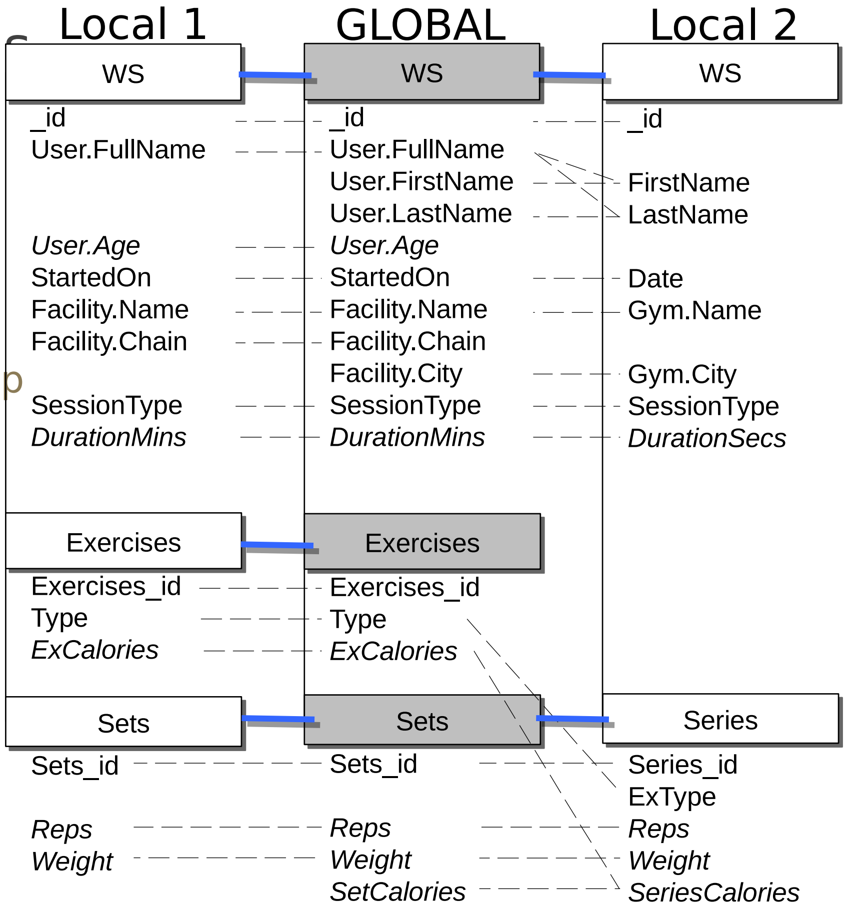
Approximate

Schema integration

Goal: integrate the local schemas to obtain a s
schema

Integration through mappings

- Primitive mappings
  - Only **exact** mappings
  - Transcoding functions required
- Array mappings
  - Define the context of primitive mapp



Approximate

Schema integration

Goal: integrate the local schemas to obtain a s
schema

1. Build a preliminary global schema as the name-based union of all local schemas (automated)
2. refine the preliminary global schema by iteratively merging matching (sets of) fields (semi-automated)

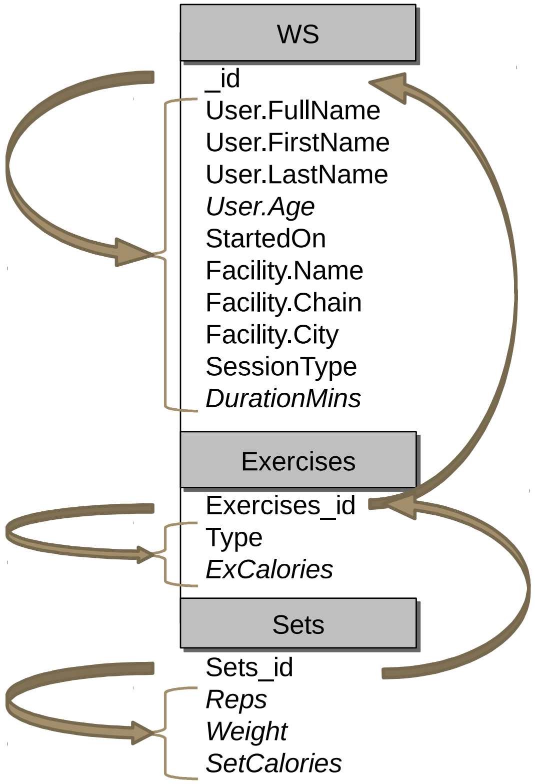
FD enrichment

Goal: give an MD view of the global schema to enable OLAP analyses

To build MD hierarchies we search for functional dependencies (FDs)

FDs can be identified

- From the schema (intensional)



Approximate OLAP

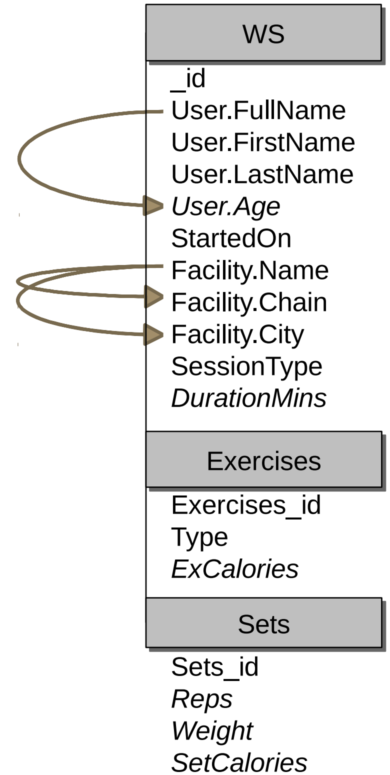
FD enrichment

Goal: give an MD view of the global schema to enable OLAP analyses

To build MD hierarchies we search for functional dependencies (FDs)

FDs can be identified

- From the schema (intensional)
- From the data (*approximate FDs*)



Approximate OLAP

FD enrichment

Goal: give an MD view of the global schema to enable OLAP analyses

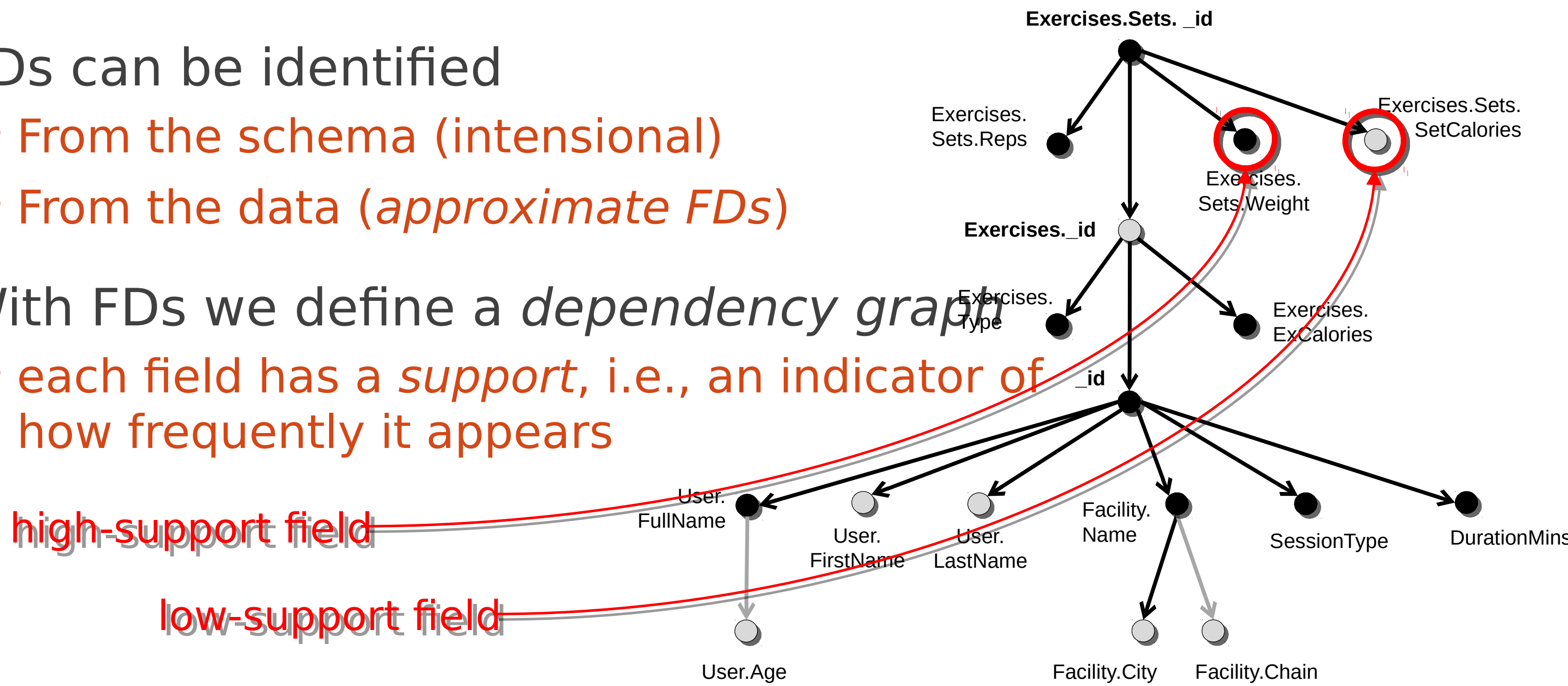
To build MD hierarchies we search for functional dependencies (FDs)

FDs can be identified

- From the schema (intensional)
- From the data (*approximate FDs*)

With FDs we define a *dependency graph*

- each field has a *support*, i.e., an indicator of how frequently it appears



Approximate OLAP

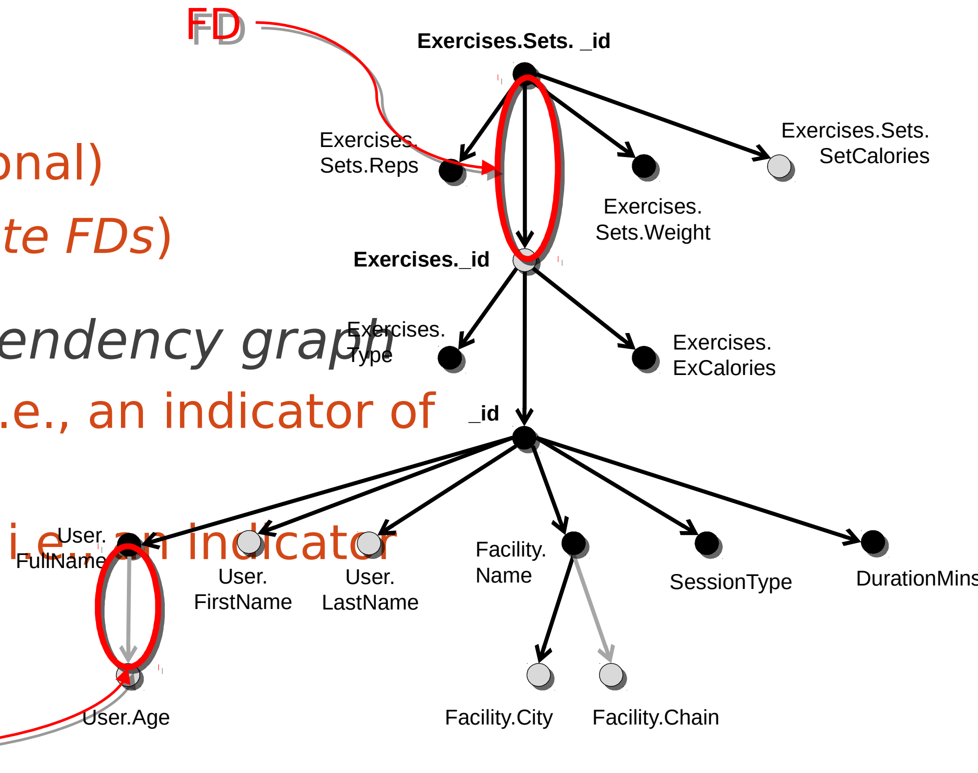
FD enrichment

Goal: give an MD view of the global schema to enable OLAP analyses

To build MD hierarchies we search for functional dependencies (FDs)

FDs can be identified
- From the schema (intensional)
- From the data (*approximate FDs*)

With FDs we define a *dependency graph*
- each field has a *support*, i.e., an indicator of how frequently it appears
- each FD has an *accuracy*, i.e., an indicator of how frequently it holds



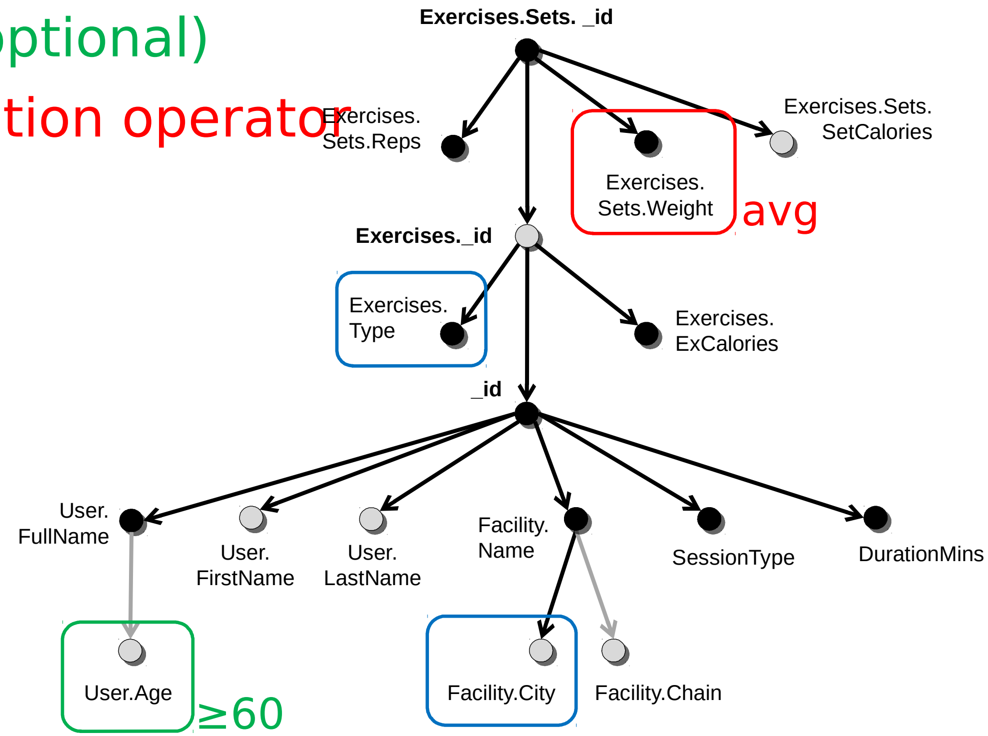
Approximate OLAP

Querying

Goal: **formulate**, execute, evaluate

OLAP query

- Group-by set (non-empty)
- Selection predicate (optional)
- Measure and aggregation operator



Approximate

Querying

Goal: **formulate**, execute, evaluate

For reformulating queries from the global schema to the local schemas we rely on the BIN framework[1]

- an approach to enable OLAP on a P2P data warehousing architecture

Compliance with the BIN framework guarantees the correctness of the approach

[1] M. Golfarelli, F. Mandreoli, W. Penzo, S. Rizzi, E. Turricchia. OLAP Query Reformulation in Peer-to-Peer Data Warehousing. Information Systems, vol. 37, n. 5, pp. 393-411, 2012

Querying

Goal: formulate, **execute**, evaluate

Execution requires to

- Translate each local query to MongoDB according to its query language
- Execute each local query
- Collect and aggregate the results

```
db.WS.aggregate({
   { $unwind:  "$Exercises" },
   { $unwind:  "$Exercises.Sets" },
   { $match:  { "User.Age":  { $gte:  60 } } }
   { $project:  {
     "Facility.City":  { $ifNull:
        ["$FacilityCity","$FacilityName"] }
     },
     "Exercises.Type":  1,
     "Exercises.Sets.Weight":  1,
     "balanced":  {
        $cond:  ["$FacilityCity",false,true]
     }
   } },
   { $group:  {
     "_id":  {
        "FacilityCity","$FacilityCity",
        "ExercisesType","$Exercises.Type",
        "balanced","$balanced"
     },
     "Exercises.Sets.Weight":  {
        $avg:  "$Exercises.Sets.Weight"
     },
     "count":  { $sum:  1 },
     "count-m":  { $sum:  {
        $cond:  ["$Exercises.Sets.Weight",1,0]
     } }
   } }
}
```

Approximate OLAP

Querying

Goal: formulate, execute, **evaluate**

We introduce indicators to evaluate the quality of an OLAP query in terms of coverage and reliability

- Completeness takes into account missing values of hierarchies
- Precision takes into account missing values of measures

To estimate these indicators *before* query execution, we resort to schema profiling

Experimental results

Efficiency

- extraction

# records	DB size	Time (standalone)	Time (cluster)
5 K	2 MB	4 sec	3 sec
50 K	20 MB	33 sec	19 sec
500 K	197 MB	6 min	3 min
5 M	1.7 GB	60 min	32 min

- enrichment (overall, 32 minutes

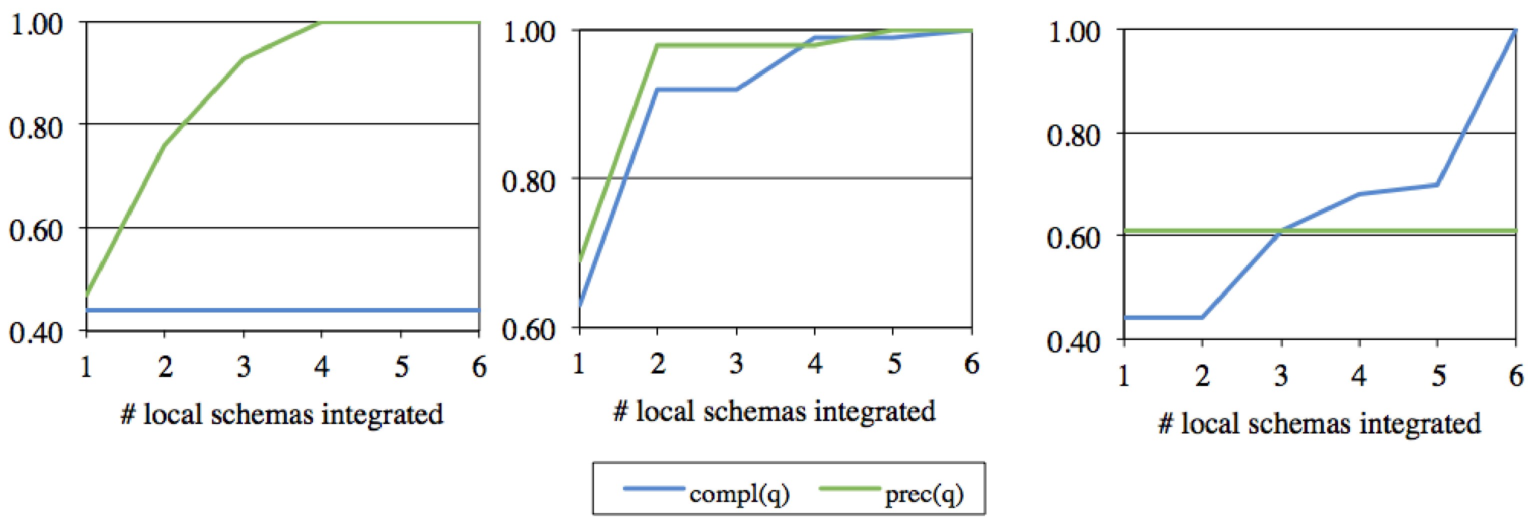
f	$\|f\|$	*Distinct count*	User.FullName	User.LastName	Facility.Name	User.FirstName	Ex.ExCalories	Ex.Sets.SetCalories	Facility.City	Ex.Sets.Weight	DurationMins	StartedOn	Facility.Chain	Ex.Type	User.Age	Ex.Sets.Reps	Facility.Country	SessionType
User.FullN	416K	5	-	12	11	11	70	110	10	108	10	10	9	104	10	96	11	10
User.LastN	228K	4	12	-	5	5	53	89	4	79	4	3	3	66	2	81	2	2
Fac.Name	2134	1	11	5	-	3	42	67	3	66	2	2	2	39	2	63	2	2
User.FirstN	1845	3	11	5	3	-	40	68	3	67	2	2	2	39	2	62	2	2
Ex.ExCal	803	33	70	53	42	40	-	68	39	67	37	38	37	39	38	67	37	37
Ex.S.SetCal	515	54	110	89	67	68	68	-	68	75	67	67	66	66	65	70	65	64
Fac.City	298	1	10	4	3	3	39	68	-	70	2	2	2	37	1	62	1	1
Ex.S.Wght	204	54	108	79	66	67	67	75	70	-	66	65	65	65	65	68	63	63
DurMins	160	1	10	4	2	2	37	67	2	66	-	1	1	36	1	60	1	1
StartedOn	122	1	10	3	2	2	38	67	2	65	1	-	1	36	1	61	1	1
Fac.Chain	103	1	9	3	2	2	37	66	2	65	1	1	-	37	1	61	1	1
Ex.Type	100	32	104	66	39	39	39	66	37	65	36	36	37	-	40	62	39	39
User.Age	70	1	10	2	2	2	38	65	1	65	1	1	1	40	-	60	1	1
Ex.S.Reps	40	53	96	81	63	62	67	70	62	68	60	61	61	62	60	-	74	73
Fac.Country	5	1	11	2	2	2	37	65	1	63	1	1	1	39	1	74	-	1
SessType	3	1	10	2	2	2	37	64	1	63	1	1	1	39	1	73	1	-

Approximate OLAP

Experimental results

Effectiveness

- query completeness and precision during a progressive integration of local schemas



Conclusion

Dealing with heterogeneity and schema variety intrinsic to document-oriented DBs is a challenge

We claim variety should be considered as a source of information wealth and shown to users together with an assessment of its impact in terms of completeness and precision

To this end, we enable OLAP queries over the collection and make users aware of the impact of schema variety through a set of indicators related to query completeness and precision

Thanks