Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Advanced Data Profiling

Efficient membership tests for Order Dependencies

WS 2023/2024

Felix Naumann, Sebastian Schmidl
Youri Kaminsky, Daniel Lindner

Design IT.
Create Knowledge.

www.hpi.de

About Us

Prof. Dr. Felix Naumann

- Data Profiling – its uses, benefits, and caveats
- Dependency Discovery Algorithms
- „Bigger Picture“

Youri Kaminsky

- Data Profiling Generalist
- Dependency Discovery Algorithms
- Metanome

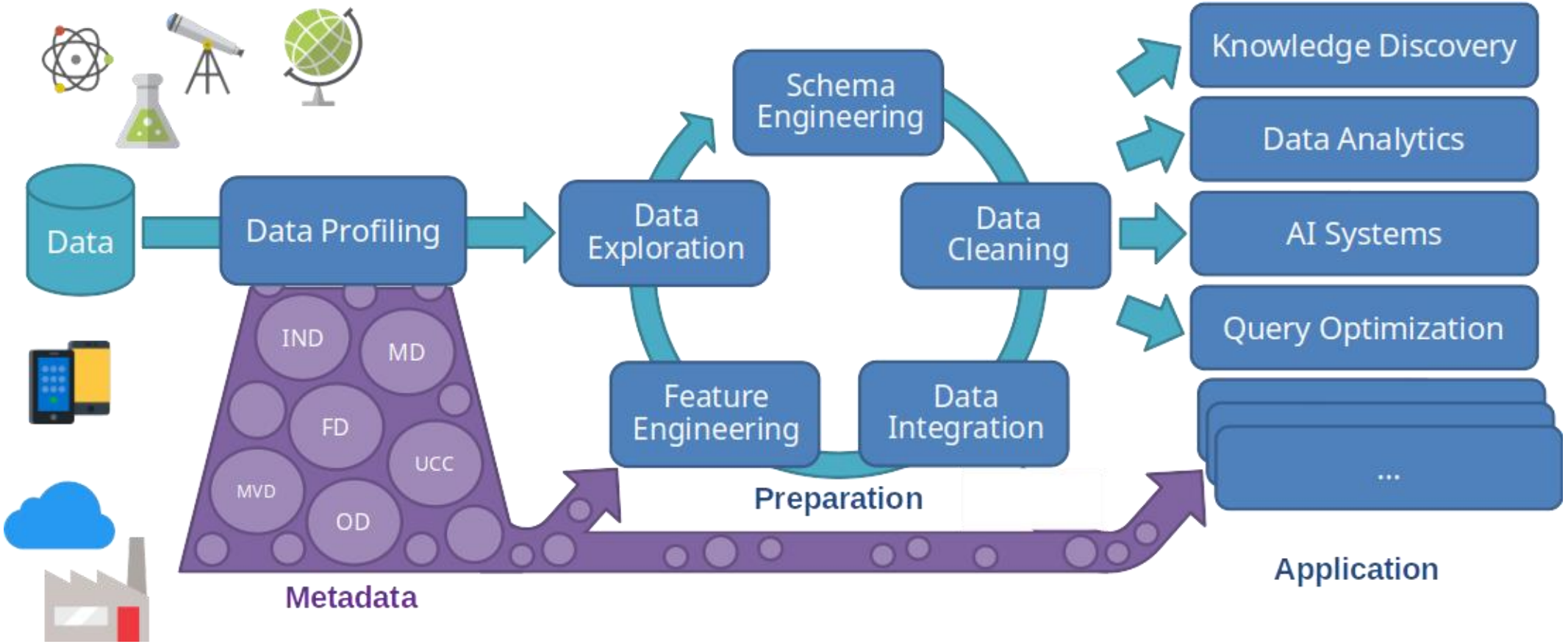
Daniel Lindner

- Application of Dependencies in DB Query Optimization
- DB internas

Sebastian Schmidl

- ODs
- Efficient Detection of ODs
- Distributed Systems

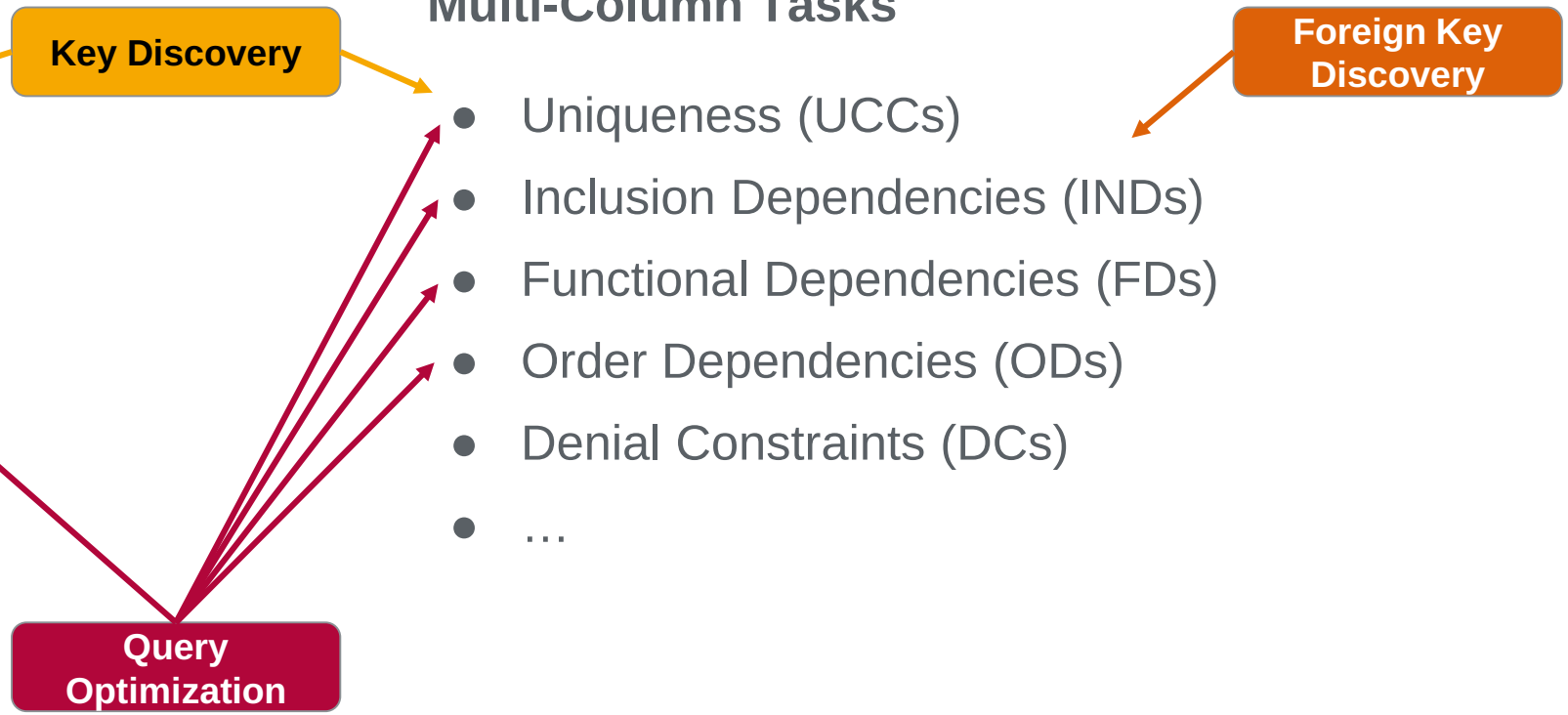
Data Profiling for Data Engineering



Classifying Data Profiling Tasks

Single-Column Tasks

- Cardinalities
- Uniqueness
- Patterns and Data Types
- Distributions
- …



Multi-Column Tasks

- Uniqueness (UCCs)
- Inclusion Dependencies (INDs)
- Functional Dependencies (FDs)
- Order Dependencies (ODs)
- Denial Constraints (DCs)
- …

Data Profiling Use Case: Query Optimization

- Unique Column Combinations (UCCs)
- Functional Dependencies (FDs)
- Order Dependencies (ODs)
- Inclusion Dependencies (INDs)

58 optimization opportunities

Application area	Unique Column Combinations (Sec. 5)	Functional Dependencies (Sec. 6)	Order Dependencies (Sec. 7)	Inclusion Dependencies (Sec. 8)
Join	• Spurious-free back-joins [127] * • Semijoin transformation [87] * • Pipeline with grouping [30, 124] † • Invisible join [2] †	• Simplification / avoidance [65] * • Complexity reduction (Sec. 6.2) * • Self-join avoidance [6] * • Plan generation [38] †	• Join avoidance [115, 117] * • Pipeline with grouping [23, 49] † • Avoid sort for sort-merge-joins [42, 100, 110] † • Attribute substitution (Sec. 7.2) † • Pipeline index scan with join [49] †	• Join elimination [24, 64] * • Substitute relations [33] † • Avoid semijoin reductions (Sec. 8.1) † • Accurate cardinalities [56] ‡
Selection	• Early abort (Sec. 5.6) * • Accurate cardinalities (Sec. 5.6) ‡	• Early abort (Sec. 6.3) * • Substitute attributes [24, 68] † • Estimations without independency assumption [25, 58, 106] ‡	• Use binary search [100] †	
Grouping & Aggregate functions	• Grouping is redundant [23] * • Accurate cardinalities (Sec. 5.6) ‡	• Reduce attributes [17, 114] *	• Simplify MIN, MAX, MEDIAN [93, 100] * • Sort-based grouping [100, 110, 115, 125, 126] †	
Projection & Distinctness	• Avoid DISTINCT [99, 100, 101] *	• Distinctness: see grouping [121] * • Simplification [29] * • Estimate projections [44] ‡	• Distinctness: See grouping †	
Sorting	• Reduce attributes (Sec. 5.6) * • Unstable sorting (Sec. 5.6) *	• Reduce attributes [24, 110, 114] *	• Reduce attributes [113, 114, 115] * • Avoid sort [49, 100] * • ORDER BY with index [115] * • Main-memory sorts [115] † • Substitute attributes [100] † • Accurate estimates (Sec. 7.4) ‡	
Set Operations	• EXCEPT to EXCEPT ALL [100] * • INTERSECT to INTERSECT ALL [60, 99] * • INTERSECT to join [99, 100] † • Accurate cardinalities (Sec. 5.6) ‡		• Order optimizations [100] †	• Simplify UNION (Sec. 8.2) * • Simplify INTERSECT (Sec. 8.2) * • Eliminate EXCEPT (Sec. 8.2) * • Accurate cardinalities (Sec. 8.2) ‡
Other	• Subquery to join [99, 100] * • Subquery sort avoidance [108] †	• Scalar subqueries [29] • Table decomposition rewrites [45]	• Correlated subqueries [100] ‡ • Sparse over dense indexes [34]	• Query folding [33, 51, 57] * • Eliminate correlated subqueries in EXISTS [83] (Sec. 8.2) *

Kossmann et al: *Data dependencies for query optimization: a survey.* VLDBJ 31(1): 1-22 (2022).

Order Dependencies (ODs)

salary ↦ tax

salary	tax
5k	1k
6k	1.5k
8k	2k
10k	3k

year ↦ CO2

year	CO2
2019	411.49
2018	408.59
2017	406.59
2016	404.28

https://www.co2.earth/annual-co2

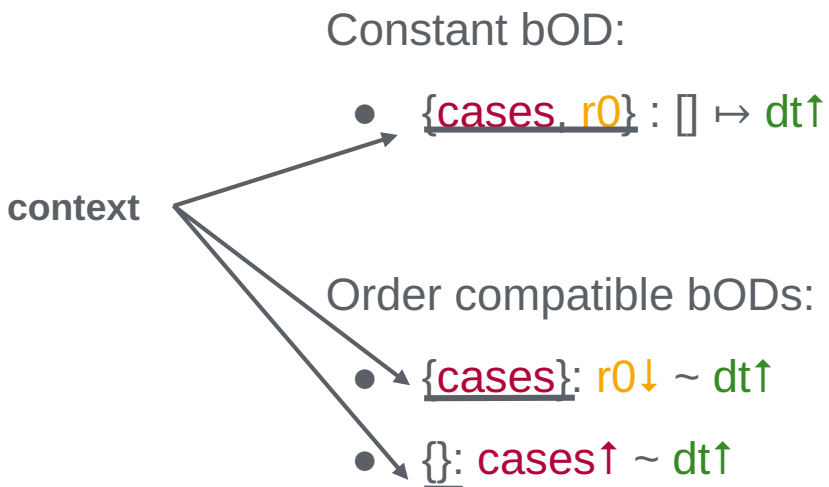
[cases↑, r0↓] ↦ [dt↑]

cases	r0	doubling time
46	1.5	4 d
57	1.8	7 d
102	1.7	10 d
102	1.4	12 d
188	1.4	14 d

- Lexicographical ordering
- Think SQL:
  `... ORDER BY cases asc, r0 desc`
  `⇔ ... ORDER BY „doubling time“ asc`
- Default order direction: asc = ↑

Set-based canonical form for ODs

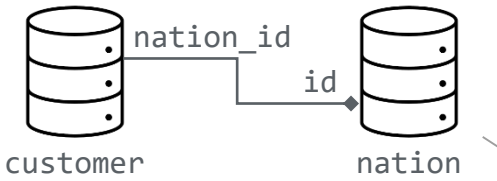
	cases	r0	doubling time
t1	46	1.5	4 d
t2	57	1.8	7 d
t3	102	1.7	10 d
t4	102	1.4	12 d
t5	188	1.4	14 d



Equivalent to:

[cases↑, r0↓] ↦ [dt↑]

Example: Join Elimination with ODs



id	region	name
1	AFRICA	Algeria
2	AFRICA	Angola
3	AFRICA	Benin
…	…	…
54	AFRICA	Zimbabwe
55	AMERICA	Argentina
…	…	…
193	OCEANIA	Vanuatu

```
    nation.region = 'AFRICA'
AND nation.name LIKE 'A%'
```

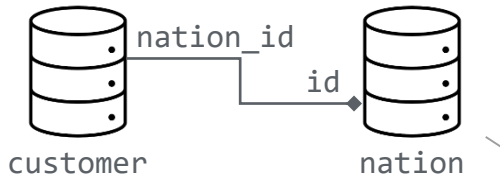
$[\text{nation.id}\uparrow] \mapsto [\text{nation.region}\uparrow, \text{nation.name}\uparrow]$

```
SELECT …
  FROM customer, nation
 WHERE customer.nation_id = nation.id
   AND nation.region = 'AFRICA'
   AND nation.name LIKE 'A%'
…;
```

```
SELECT …
  FROM customer,
       (SELECT min(nation.id) as key_min,
               max(nation.id) as key_max
          FROM nation
         WHERE nation.region = 'AFRICA'
           AND nation.name LIKE 'A%') as n2,
 WHERE nation_id BETWEEN n2.key_min AND n2.key_max
…;
```

19 TPC-DS queries match this rewrite with avg. **runtime improvement of 32%**.

Example: Join Elimination with ODs



id	region	name
1	AFRICA	Algeria
2	AFRICA	Angola
3	AFRICA	Benin
…	…	…
54	AFRICA	Zimbabwe
55	AMERICA	Argentina
…	…	…
193	OCEANIA	Vanuatu

```
    nation.region = 'AFRICA'
AND nation.name LIKE 'A%'
```

[nation.id↑] ↦ [nation.region↑, nation.name↑]?

```
{name}: [] ↦ region
{}: id↑ ~ region↑
{name}: [] ↦ id
{id}: [] ↦ name
{id}: [] ↦ region
{region}: id↑ ~ name↑
```

Dataset	|R|	|r|	Size[a]	#FDs	#bODs
iris-sub.csv	5	150	4	4	7
chess-sub.csv	7	28,056	519	1	0
imdb-sub.csv	8	7,734,683	628,014	8	26
abalone-sub.csv	9	4177	187	137	318
bridges-sub.csv	13	108	6	142	1097
dblp-sub.csv	13	8 224 309	1,237,373	31	664
adult-sub.csv	15	32,561	3527	78	1140
tpch-sub.csv	16	6001 215	736,193	3984	13,760
letter-sub.csv	17	20,000	696	61	2202
ncvoter-sub.csv	19	999,999	112,578	572	4362
hepatitis-sub.csv	20	155	7	8 250	54,766
flight-long-sub.csv	21	499,999	71,032	290	2253
horse-sub.csv	29	300	25	128,727	2,231,321
flight-sub.csv	30	1000	135	5976	40,740
fd-reduced-sub.csv	30	250,000	69,580	89,571	742
plista-narrow.csv	35	1001	193	4467	54,921
plista-sub.csv	63	1001	575	≥ 32,404	≥ 313,296

Efficiently inferring a list-based OD's validity based on a large set of set-based ODs is not trivial!

Schmidl & Papenbrock: *Efficient distributed discovery of bidirectional order dependencies*. VLDBJ 31(1): 49-74 (2022).

Goals and Tasks

- Learn about the research area of data profiling
- Read and understand scientific papers
- Craft a novel solution to the problem of efficient OD membership tests
- Run experiments and evaluate results
- Present results in written and oral form

How do we efficiently check whether a given list-based OD can be derived from the set of minimal set-based ODs?

Organization

General

- Project seminar for master students
- Language: English
- ECTS: 6
- Maximum number of participants: 8 (groups of two)
- Weekly meetings: **Thursdays 11:00 in F-E.06**

German is possible

Requirements

- Prior knowledge in data profiling (e.g., Data Profiling lecture)
- Good programming skills in a major programming language

We will give a very brief introduction into the main concepts in the next meeting

Grading

- 35 % Approach
- 35 % Written report
- 30 % Presentations

10-page single-column report per group (e.g., LNI-template)

We can support you in:

- Python
- Java
- Scala
- C++

1. Goal & Idea
2. Implementation & Experiment Design

Timeline

Date	Topic
2023-10-19 (always in F-E.06)	Seminar introduction
2023-10-26	Introduction to order dependencies
2023-11-02	**Teams and Topics**
2023-11-09	*(no meeting)*
2023-11-16 – 2023-12-07	Weekly meetings and progress reports
2023-12-14	**Presentations 1** (Goal & Idea)
2023-12-21	*(no meeting)?*
2023-12-28	Christmas break
2024-01-04	Christmas break
2024-01-11 – 2024-02-01	Weekly meetings and progress reports
2024-02-08	**Presentations 2** (Implementation & Experiment Setup)
2024-03-10 ?	**Submission deadline**

Literature

Data Profiling

- *Data Profiling - Synthesis Lectures on Data Management.* Ziawasch Abedjan, Lukasz Golab, Felix Naumann, Thorsten Papenbrock, Morgan Claypool, 2019. https://link.springer.com/book/10.1007/978-3-031-01865-7

Functional Dependencies

- **Armstrong Axioms**: Dependency Structures of Data Base Relationships. William Ward Armstrong (IFIP Congress 1974). https://iremi.univ-reunion.fr/IMG/pdf/armstrong.pdf

Order Dependencies

- Formal definitions and axioms of **list-based form**: *Fundamentals of order dependencies*. Jaroslaw Szlichta, Parke Godfrey, Jarek Gryz (PVLDB 2012) https://doi.org/10.14778/2350229.2350241
- Formal definitions, axioms, and conversions of **set-based form**: *Effective and complete discovery of bidirectional order dependencies via set-based axioms*. Jaroslaw Szlichta, Parke Godfrey, Lukasz Golab, Mehdi Karger, Divesh Srivastava (The VLDB Journal 2018) https://doi.org/10.1007/s00778-018-0510-0

Literature

Membership Tests for FDs

- *An algorithmic approach to normalization of relational database schemas*. Philip A. Bernstein, Catriel Beeri (Technical Report CSRG-73, University of Toronto 1976) https://ia800105.us.archive.org/24/items/technicalreportc73univ/technicalreportc73univ.pdf
  - Part II, Sect. 1: Introduction of “The Membership Problem”
  - Part II, Sect. 3: **Linear algorithm for the membership problem**
- Corresponding paper: *Computational Problems Related to the Design of Normal Form Relational Schemas*. Catriel Beeri, Philip A. Bernstein (ACM Transactions on Database Systems 1979) https://doi.org/10.1145/320064.320066
- CHASE algorithm: *Principles of Database and Knowledge-Base Systems*. Jeffrey D. Ullman. Volume I. Principles of computer science series 14, Computer Science Press 1988

Implication for FDs

- *Foundations of databases.* Abiteboul, Serge, Richard Hull, and Victor Vianu. Vol. 8. Addison-Wesley, 1995. http://webdam.inria.fr/Alice/