Scaling out the Discovery of Inclusion Dependencies
BTW 2015, Hamburg, Germany

Sebastian Kruse, Thorsten Papenbrock, Felix Naumann
Research Assistant
Hasso Plattner Institute, Potsdam, Germany

Examples

Customers

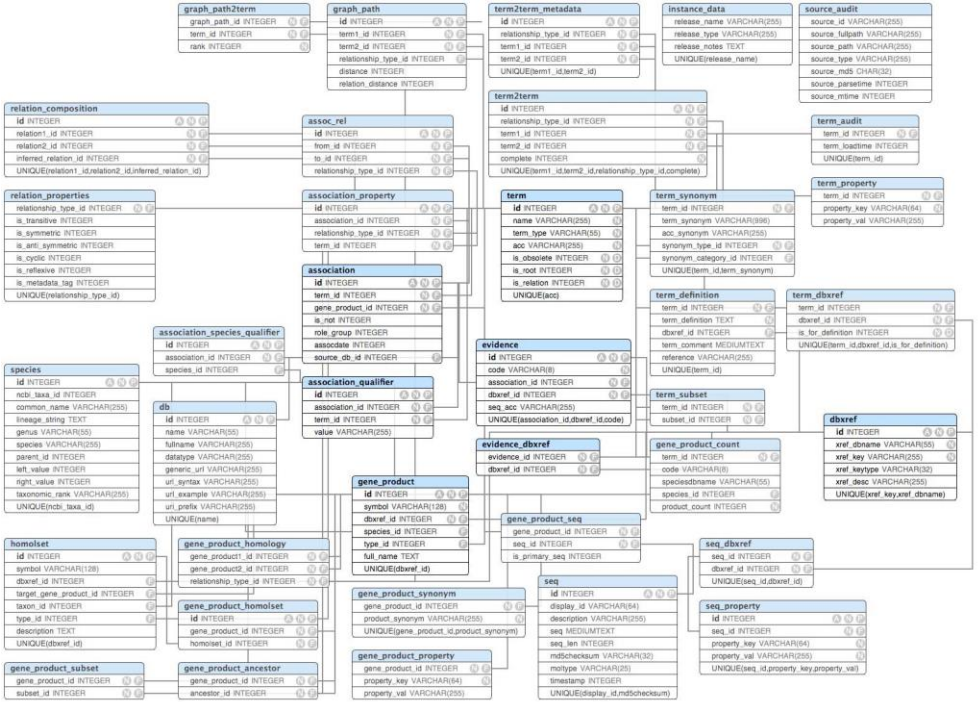
ID	Name	Address
1	Tanja Jager	Marseiller Str. 12
2	Sandra Möller	Flughafenstr. 63
3	Dennis Eberhart	Sonnenallee 19
4	Barbara Pabst	Ziegelstr. 76
5	Thorsten Mauer	Güntzelstr. 90

Orders

Customer	Item	Quantity
3	CK-242-1	1
3	DF-098-7	1
3	KE-883-6	1
1	LM-437-2	2
5	PE-383-5	1

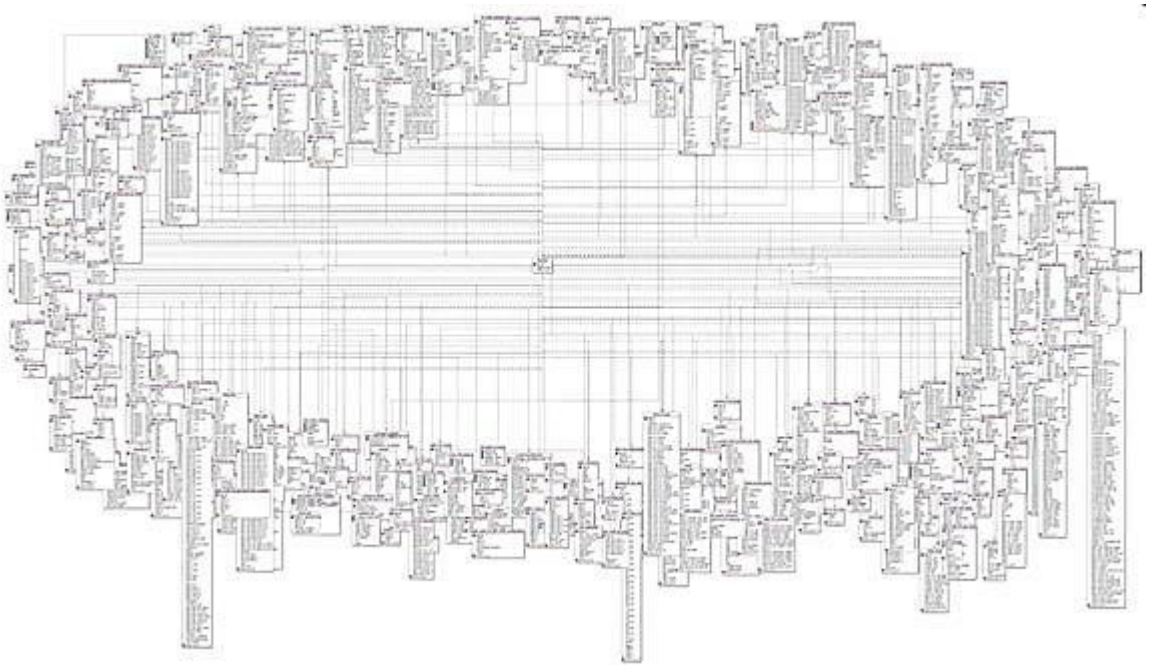
Customer ⊆ ID

Examples



http://geneontology.org/sites/default/files/public/diag-godb-er.jpg

Example



http://www.ibm.com/developerworks/data/library/techarticle/dm-1109proteindatadb2purexml/pdb_scheme_large.jpg

Agenda

1. **Discovering Inclusion Dependencies**
2. Related Work
3. SINDY: A distributed discovery algorithm
4. Evaluation
5. Conclusions

Agenda

1. Discovering Inclusion Dependencies
2. **Related Work**
3. SINDY: A distributed discovery algorithm
4. Evaluation
5. Conclusions

MIND

- Fabien De Marchi, Stéphan Lopes, and Jean-Marc Petit. Unary and n-ary inclusion dependency discovery in relational databases. *Journal of Intelligent Information Systems*, 32:53–73, 2009.

ID	Name	Address
1	Tanja Jager	Marseiller Str. 1
2	Sandra Möller	Flughafenstr. 63
3	Dennis Eberhart	Sonnenallee 19
4	Barbara Pabst	Ziegelstr. 76
5	Thorsten Mauer	Güntzelstr. 90

Customer	Item	Quantity
3	CK-242-1	1
3	DF-098-7	1
3	KE-883-6	1
1	LM-437-2	2
5	PE-383-5	1

MIND

Value	Attributes
1	ID, Customer, Quantity
Tanja Jager	Name
Marseiller Str. 12	Address
2	ID, Quantity
Sandra Möller	Name
Flughafenstr. 63	Address
…	…

Intersection **ID, Quantity**

Quantity ⊆ ID

Quantity ⊆ Quantity

SPIDER

- Jana Bauckmann, Ulf Leser, and Felix Naumann. Efficiently Computing Inclusion Dependencies for Schema Discovery. In *ICDE Workshops*, 2006.

ID	Name	Address
1	Tanja Jager	Marseiller Str. 1
2	Sandra Möller	Flughafenstr. 63
3	Dennis Eberhart	Sonnenallee 19
4	Barbara Pabst	Ziegelstr. 76
5	Thorsten Mauer	Güntzelstr. 90

Customer	Item	Quantity
3	CK-242-1	1
3	DF-098-7	1
3	KE-883-6	1
1	LM-437-2	2
5	PE-383-5	1

SPIDER

- Jana Bauckmann, Ulf Leser, and Felix Naumann. Efficiently Computing Inclusion Dependencies for Schema Discovery. In *ICDE Workshops*, 2006.

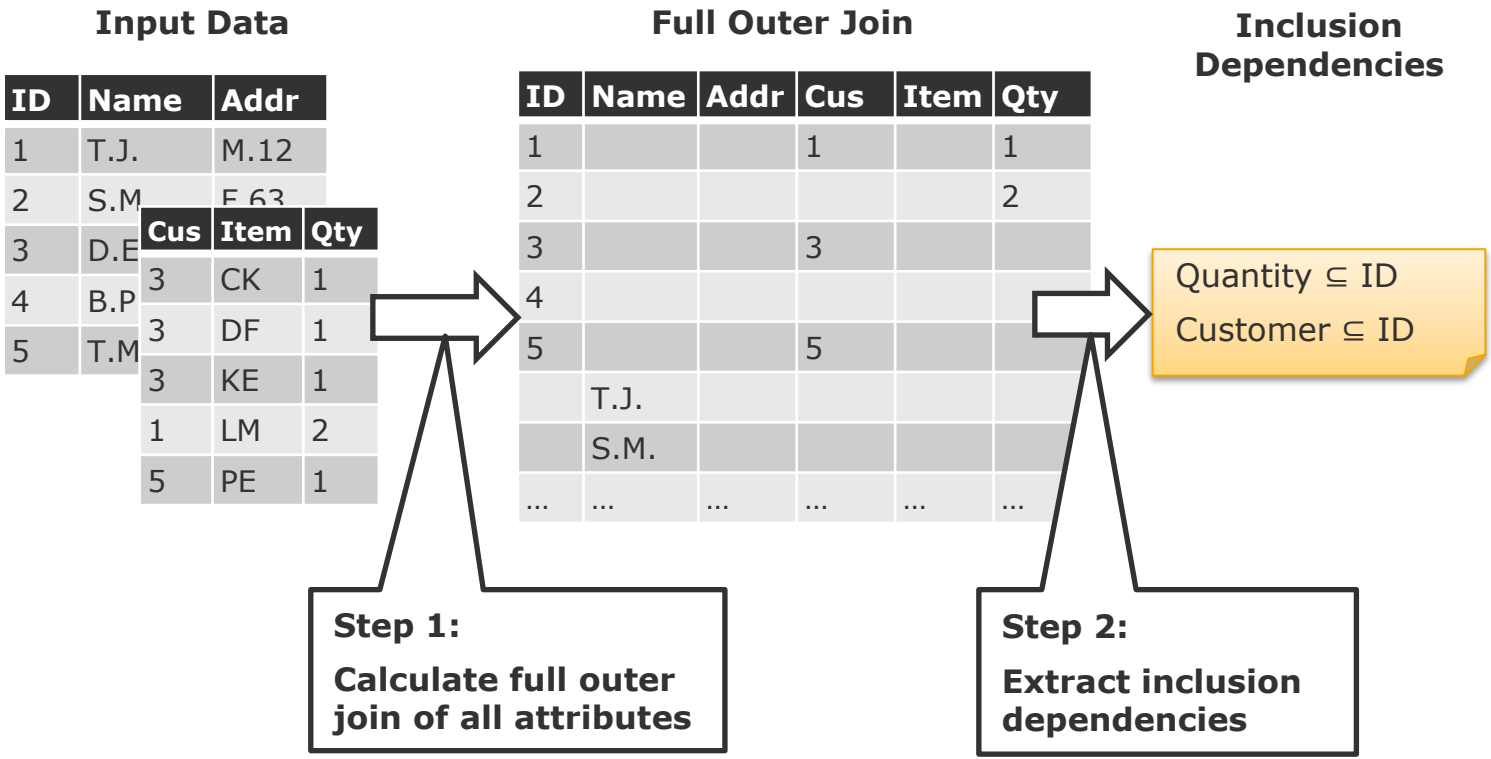
ID	Name
1	Barbara Pabst
2	Dennis Eberhart
3	Sandra Möller
4	Tanja Jager
5	Thorsten Mauer

Customer
1
3
5

Item
CK-242-1
DF-098-7
KE-883-6
LM-437-2
PE-383-5

Quantity
1
2

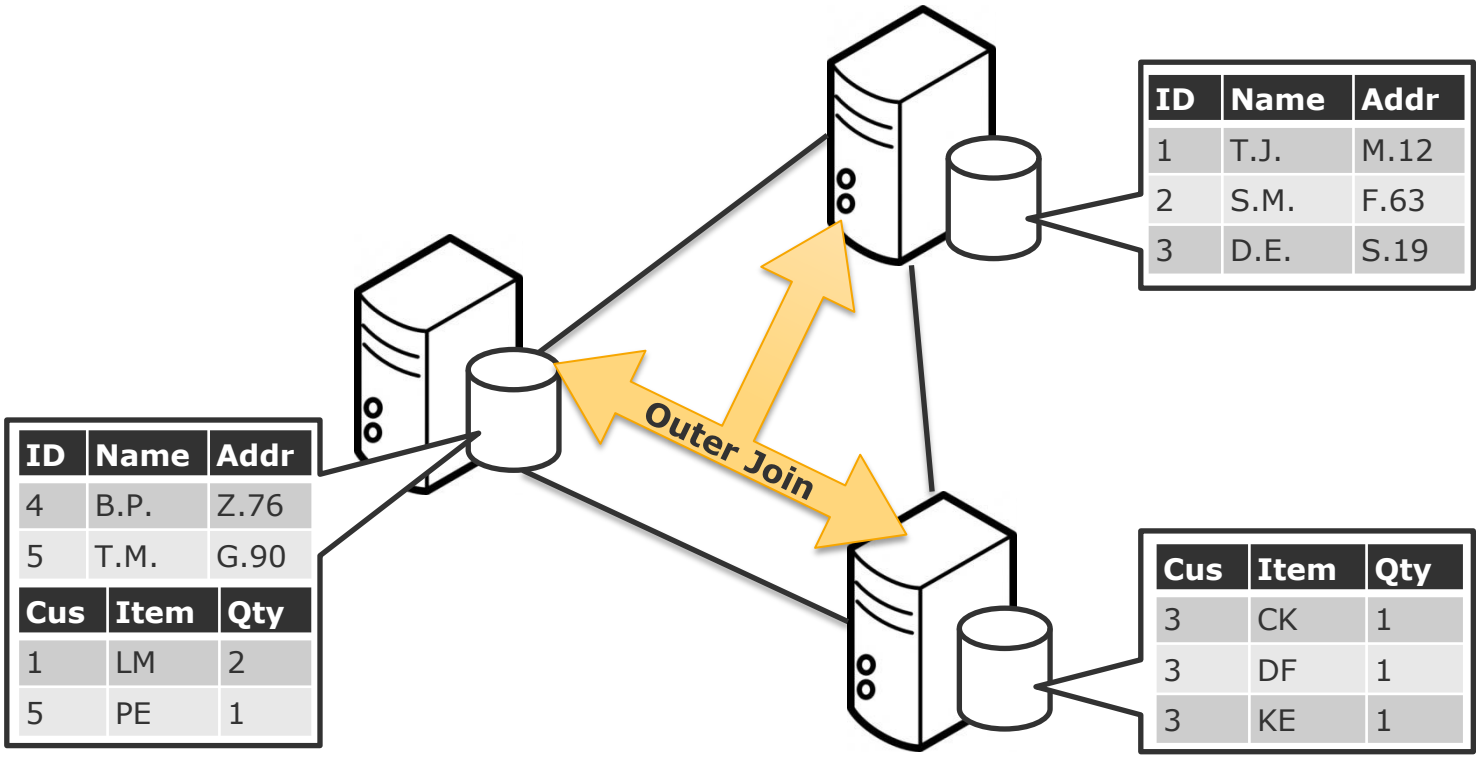
Common proceeding



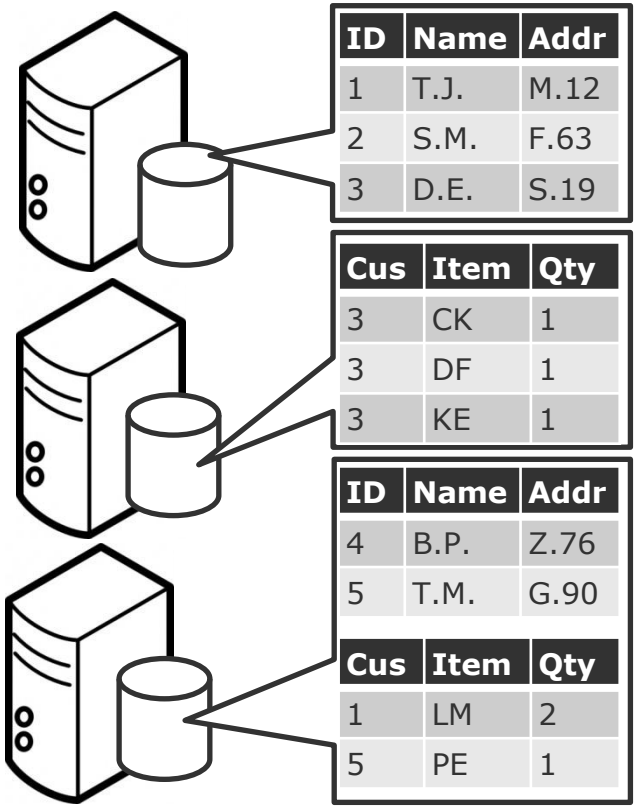
Agenda

1. Discovering Inclusion Dependencies
2. Related Work
3. **SINDY: A distributed discovery algorithm**
4. Evaluation
5. Conclusions

Distributed setting



Perform full outer join



ID	Name	Addr
1	T.J.	M.12
2	S.M.	F.63
3	D.E.	S.19

Cus	Item	Qty
3	CK	1
3	DF	1
3	KE	1

ID	Name	Addr
4	B.P.	Z.76
5	T.M.	G.90

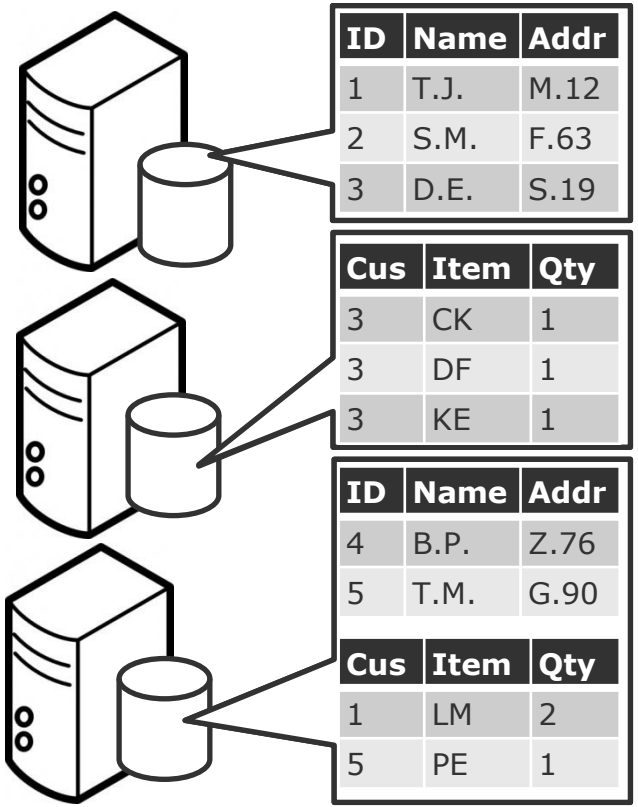
Cus	Item	Qty
1	LM	2
5	PE	1

1	ID	T.J.	Name	M.12	Addr
2	ID	S.M.	Name	F.63	Addr
3	ID	D.E.	Name	S.19	Addr

3	Cus	CK	Item	1	Qty
3	Cus	DF	Item	1	Qty
3	Cus	KE	Item	1	Qty

4	ID	B.P.	Name	Z.76	Addr		
5	ID	T.M.	Name	G.90	Addr	1	Cus, Qty
1	Cus	LM	Item	2	Qty	5	Cus, ID
5	Cus	PE	Item	1	Qty		

Perform full outer join



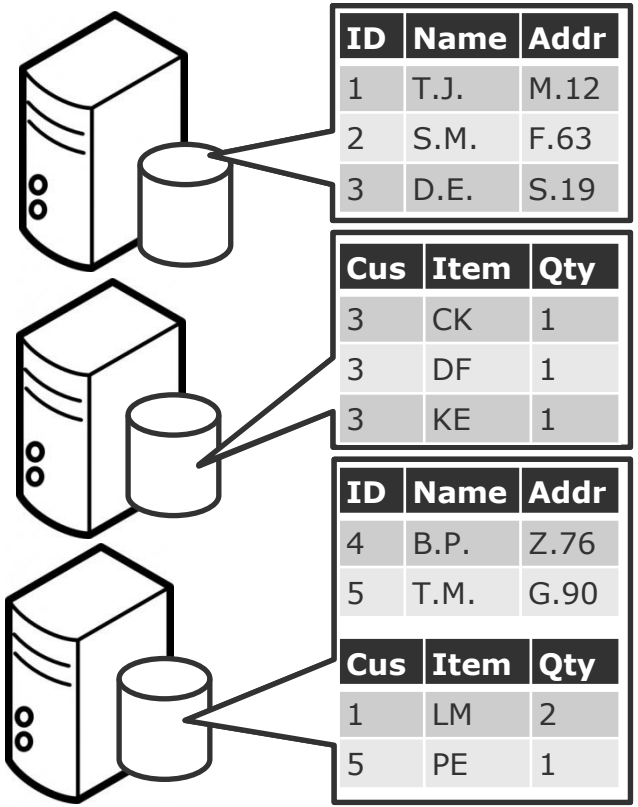
ID	Name	Addr
1	T.J.	M.12
2	S.M.	F.63
3	D.E.	S.19

Cus	Item	Qty
3	CK	1
3	DF	1
3	KE	1

ID	Name	Addr
4	B.P.	Z.76
5	T.M.	G.90

Cus	Item	Qty
1	LM	2
5	PE	1

Perform full outer join

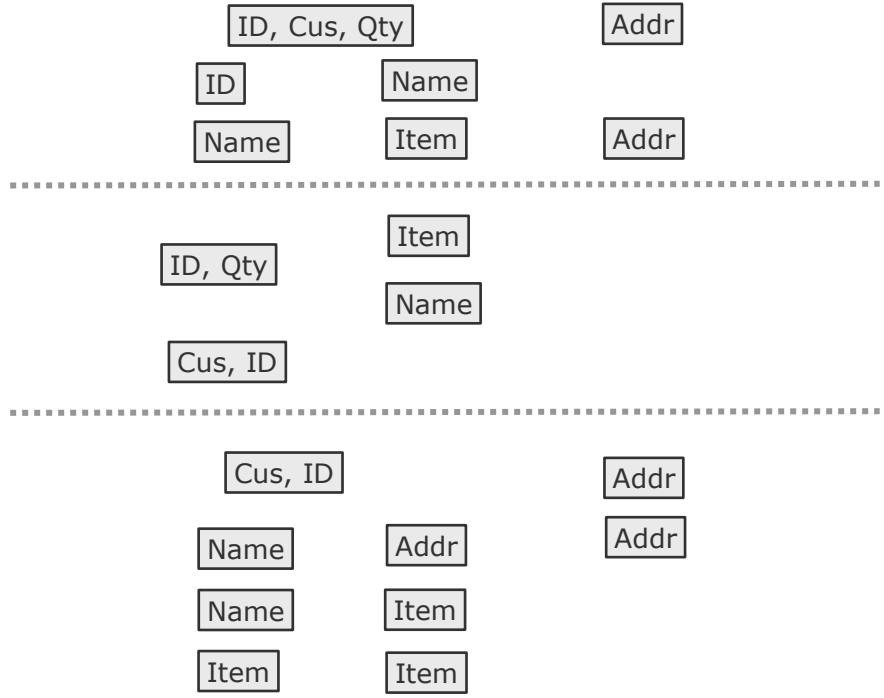


ID	Name	Addr
1	T.J.	M.12
2	S.M.	F.63
3	D.E.	S.19

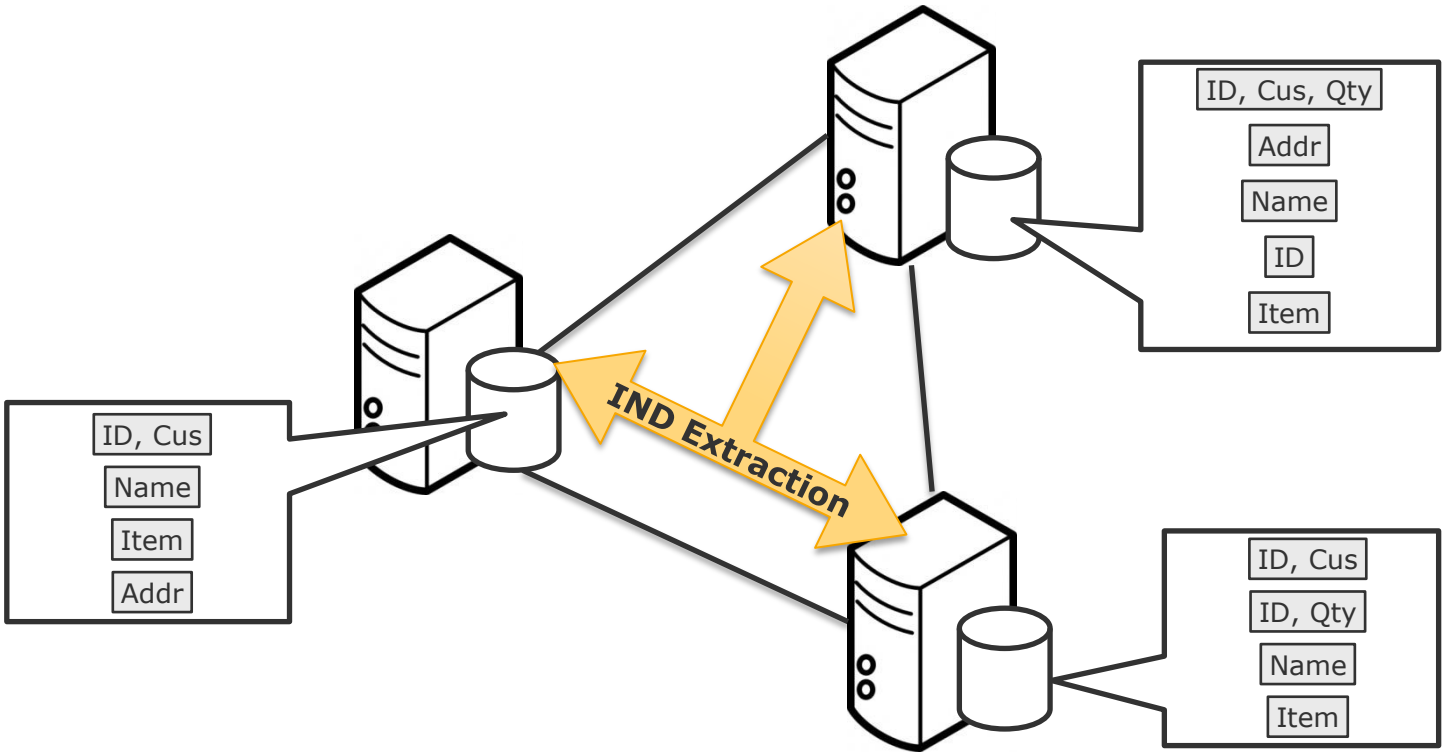
Cus	Item	Qty
3	CK	1
3	DF	1
3	KE	1

ID	Name	Addr
4	B.P.	Z.76
5	T.M.	G.90

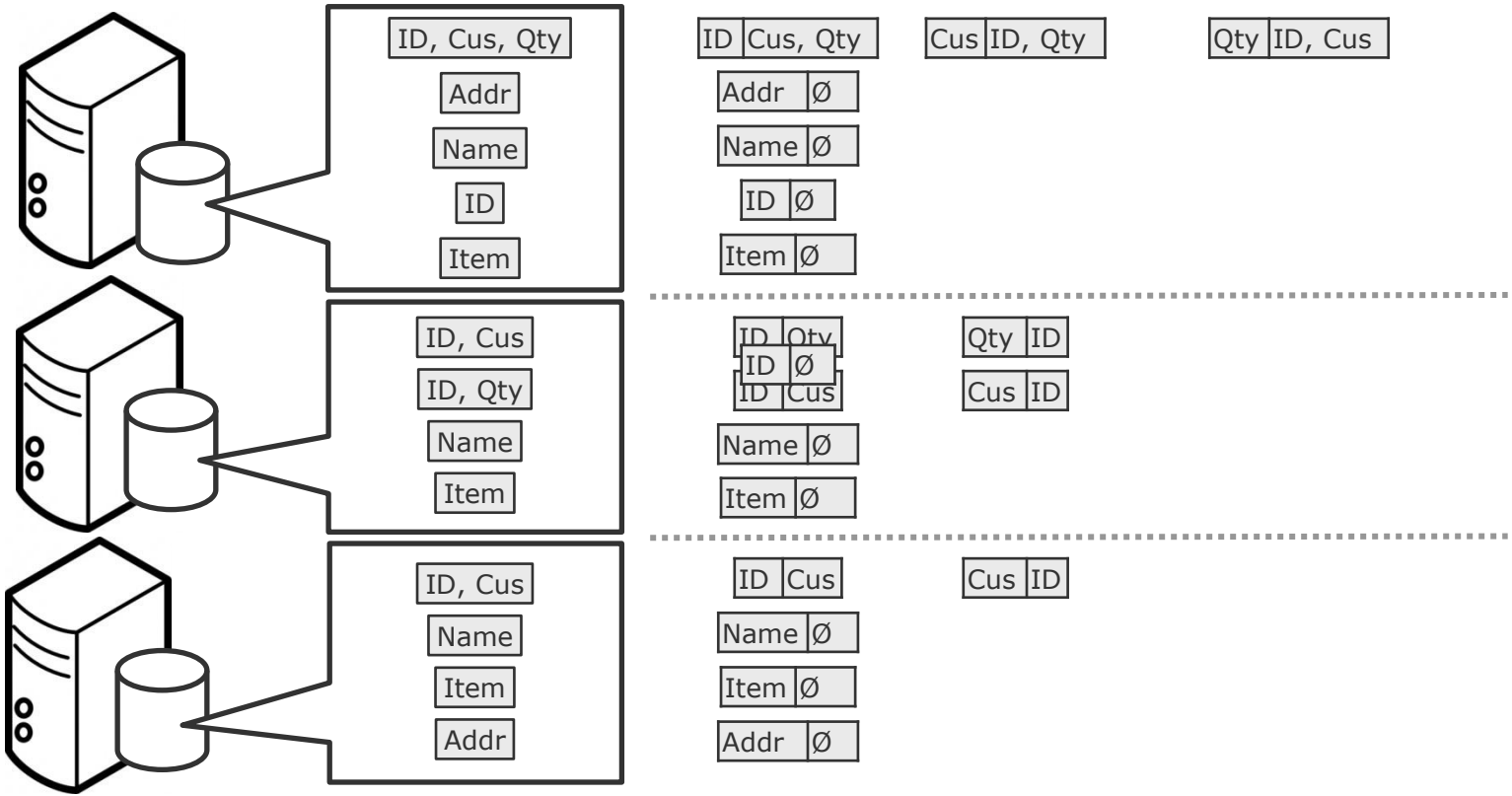
Cus	Item	Qty
1	LM	2
5	PE	1



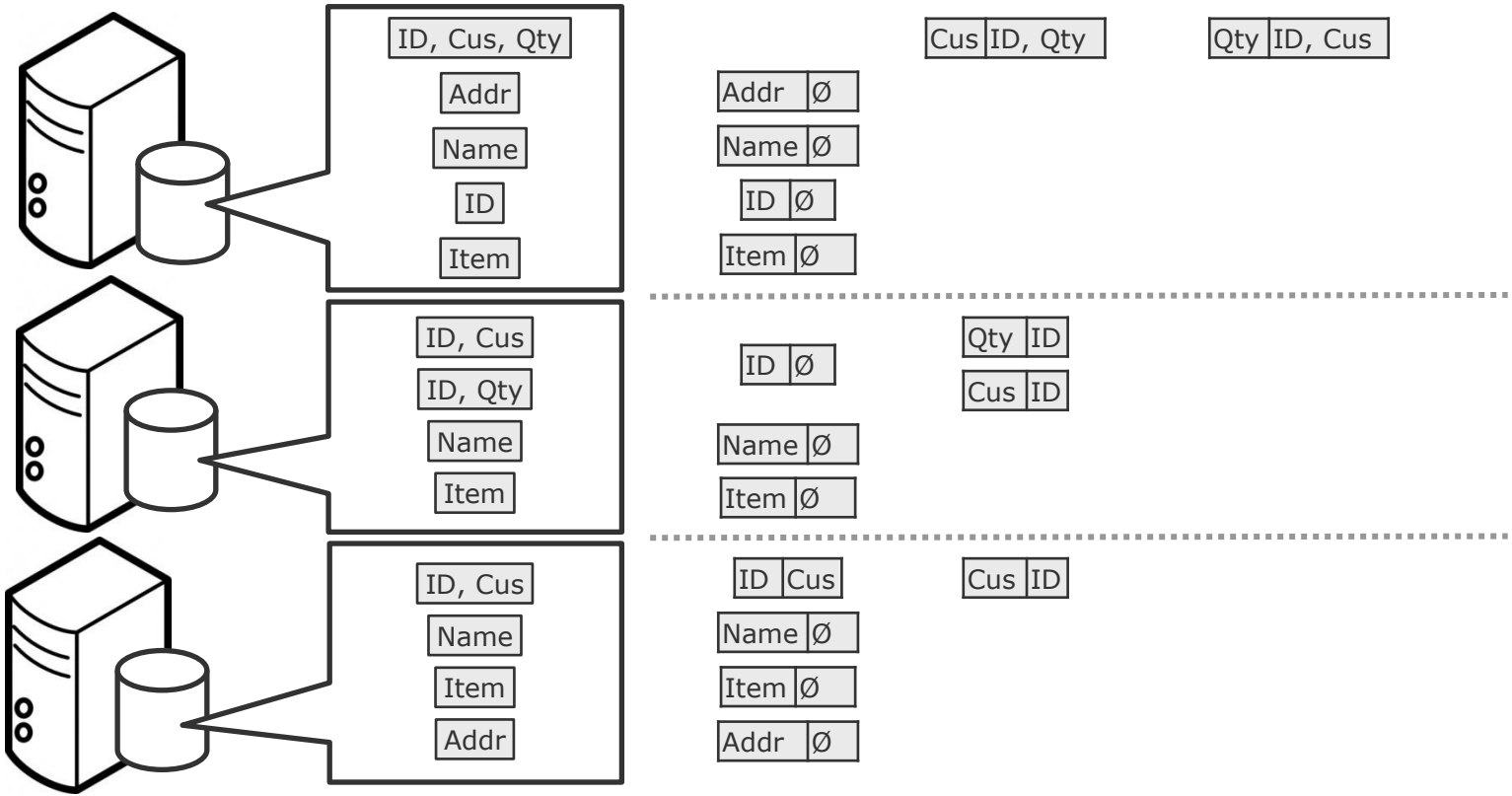
Distributed join product



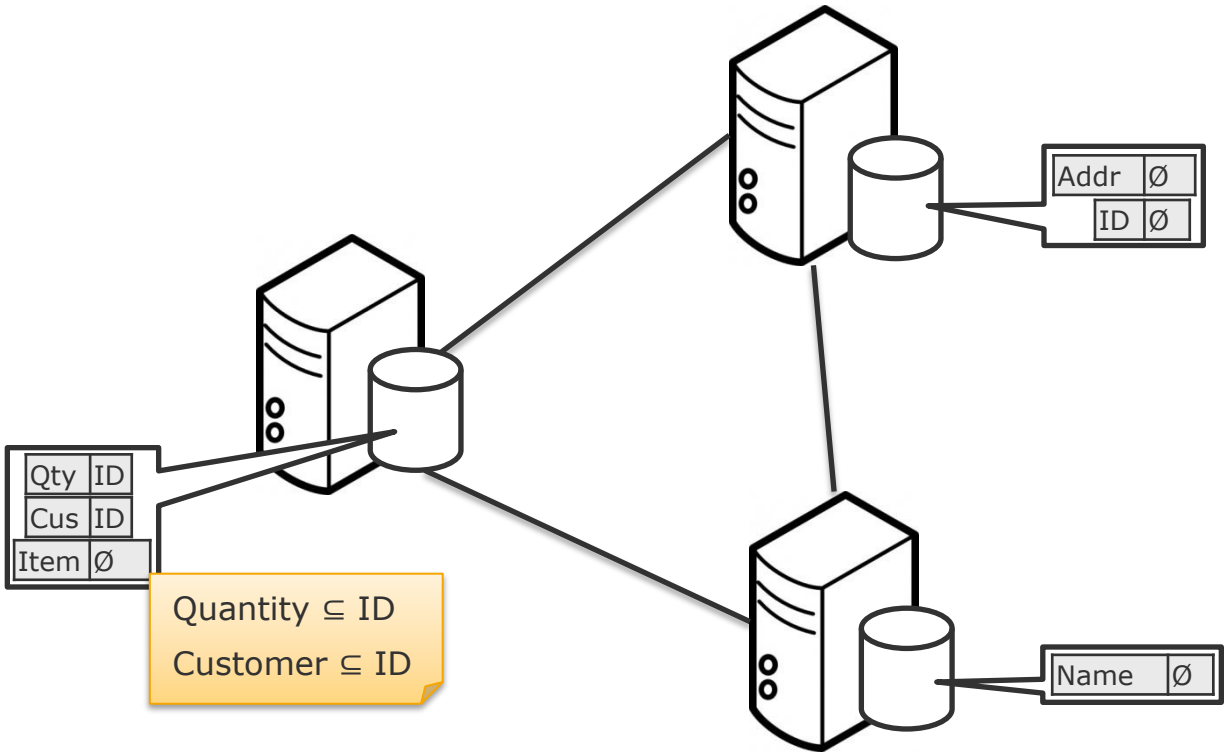
Evaluate full outer join



Evaluate full outer join



Distributed inclusion dependencies



Variants

- Inclusion dependencies on combinations of columns (aka n-ary INDs)
  - Adaption: Create cells for combinations of values
  - Powerful in combination with apriori-like proceeding

- Partial inclusion dependencies
  - Adaption: aggregate IND candidates with multiset union instead of intersection
  - Compare with number of distinct values of dependent column

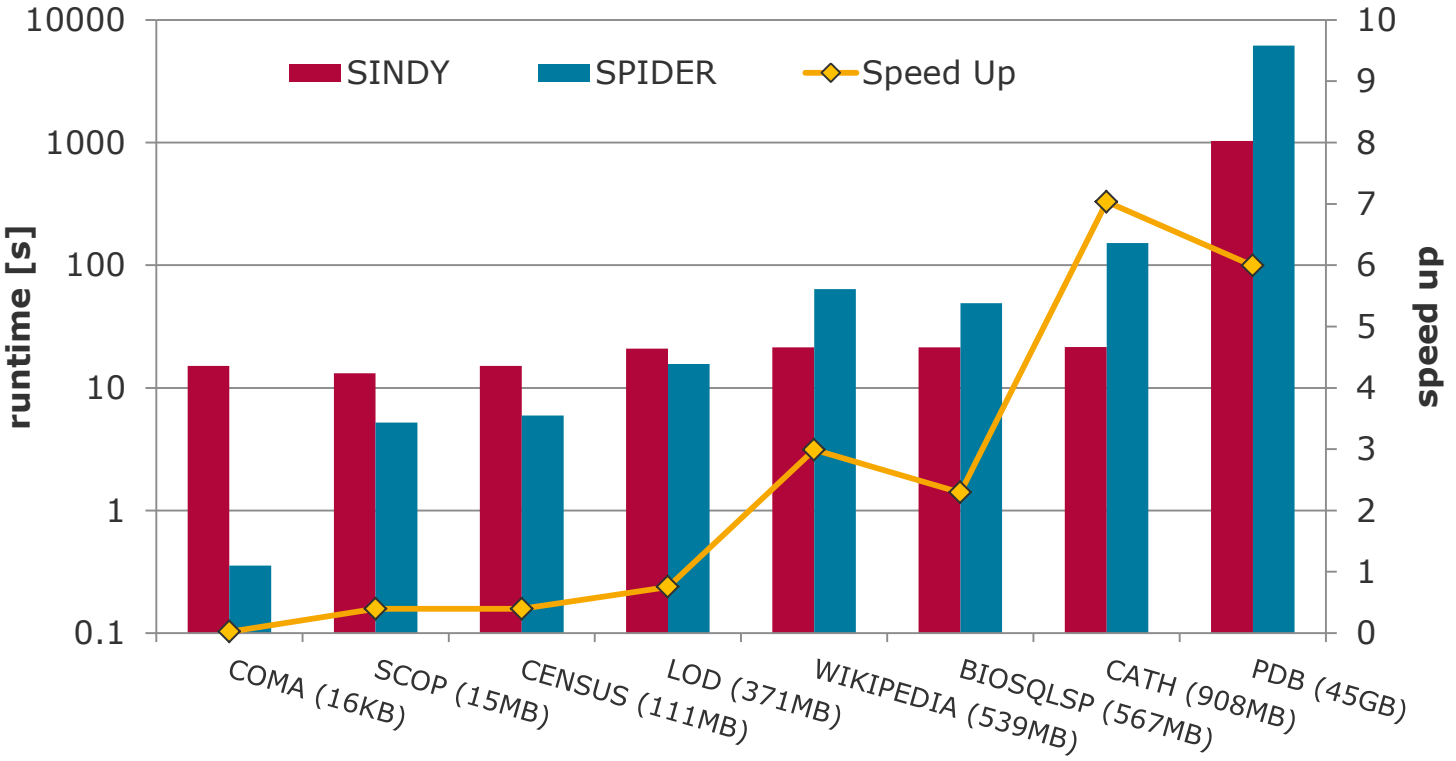
Agenda

1. Discovering Inclusion Dependencies
2. Related Work
3. SINDY: A distributed discovery algorithm
4. **Evaluation**
5. Conclusions

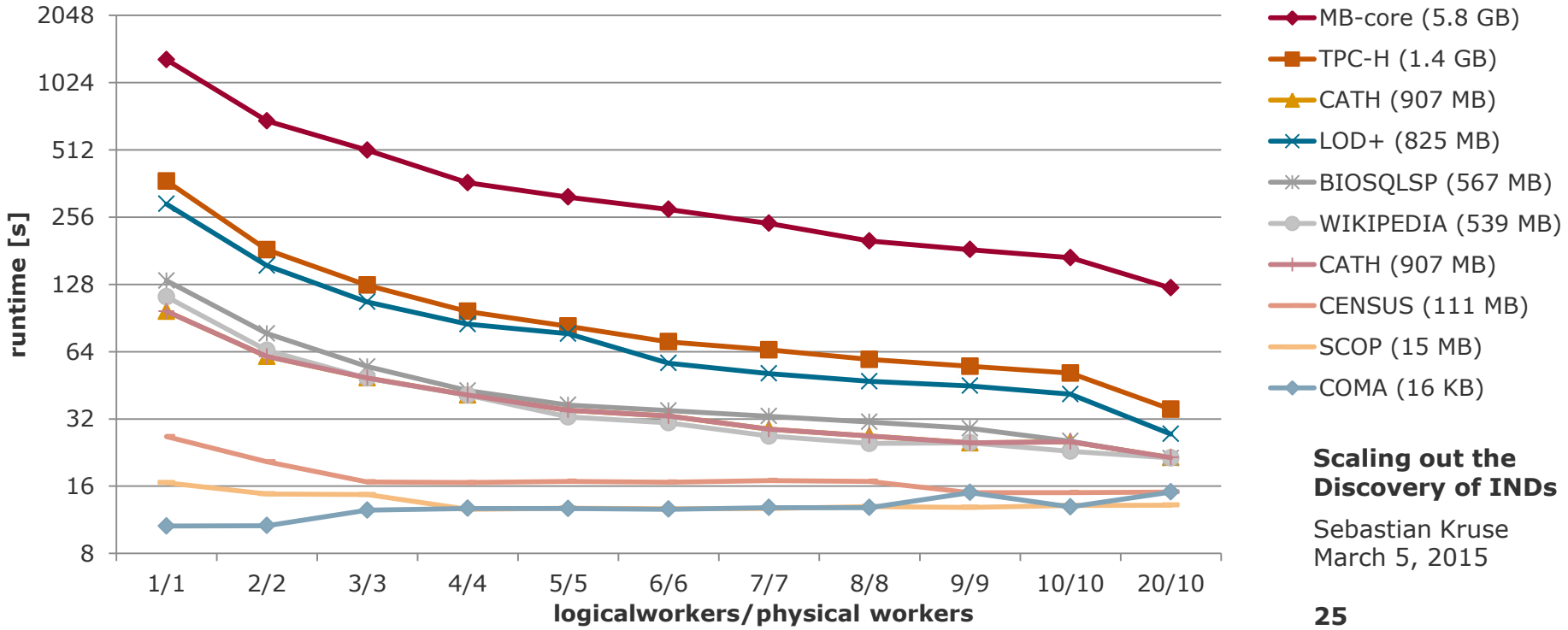
Experimental setup

- Cluster Setup
  - 1 master node (Intel Xeon @ 2x2.67 GHz, 8 GiB RAM)
  - 10 worker nodes (Intel Core 2 Duo @ 2x2.6 GHz, 8 GiB RAM)
  - Apache HDFS 2.2, Apache Flink 0.6.2
- Single node (for SPIDER)
  - Intel Xeon @ 8x2GHz, 128 GiB RAM, RAID-1
- Datasets
  - Relational datasets from different domains
  - 16 KB to 44.9 GB

Performance comparison with SPIDER



Scale-Out Behavior



Agenda

1. Discovering Inclusion Dependencies
2. Related Work
3. SINDY: A distributed discovery algorithm
4. Evaluation
5. **Conclusions**

Conclusions

- Presented new distributed IND discovery algorithm
  - Applicable for unary, n-ary, and partial inclusion dependencies
  - Consists of full outer join calculation and extraction phase
  - Scales well on large datasets

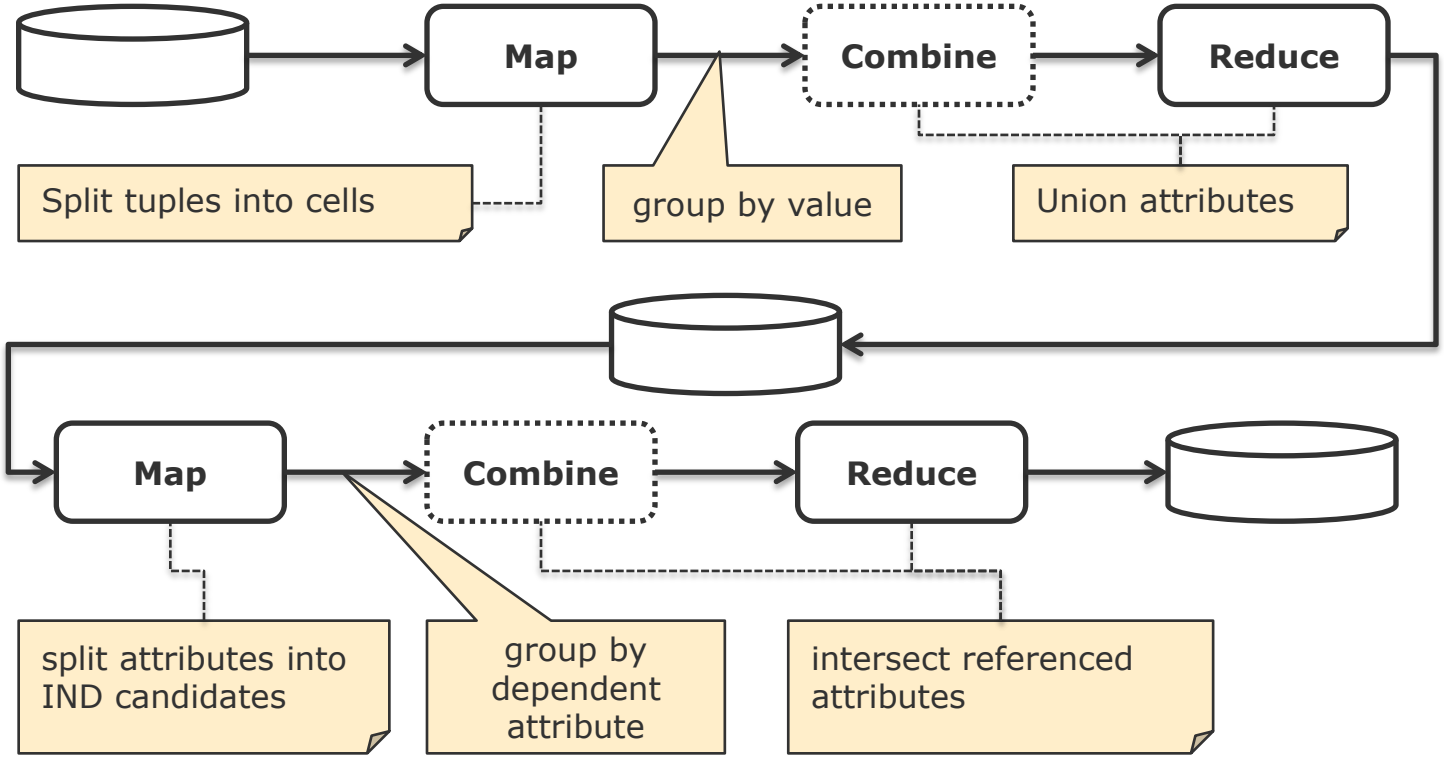
- Open questions
  - How can one continuously maintain inclusion dependencies?
  - How can the algorithm be applied to similar problems, e.g., RDF data?

Scaling out the Discovery of Inclusion Dependencies
BTW 2015, Hamburg, Germany

Sebastian Kruse, Thorsten Papenbrock, Felix Naumann
Research Assistant
Hasso Plattner Institute, Potsdam, Germany

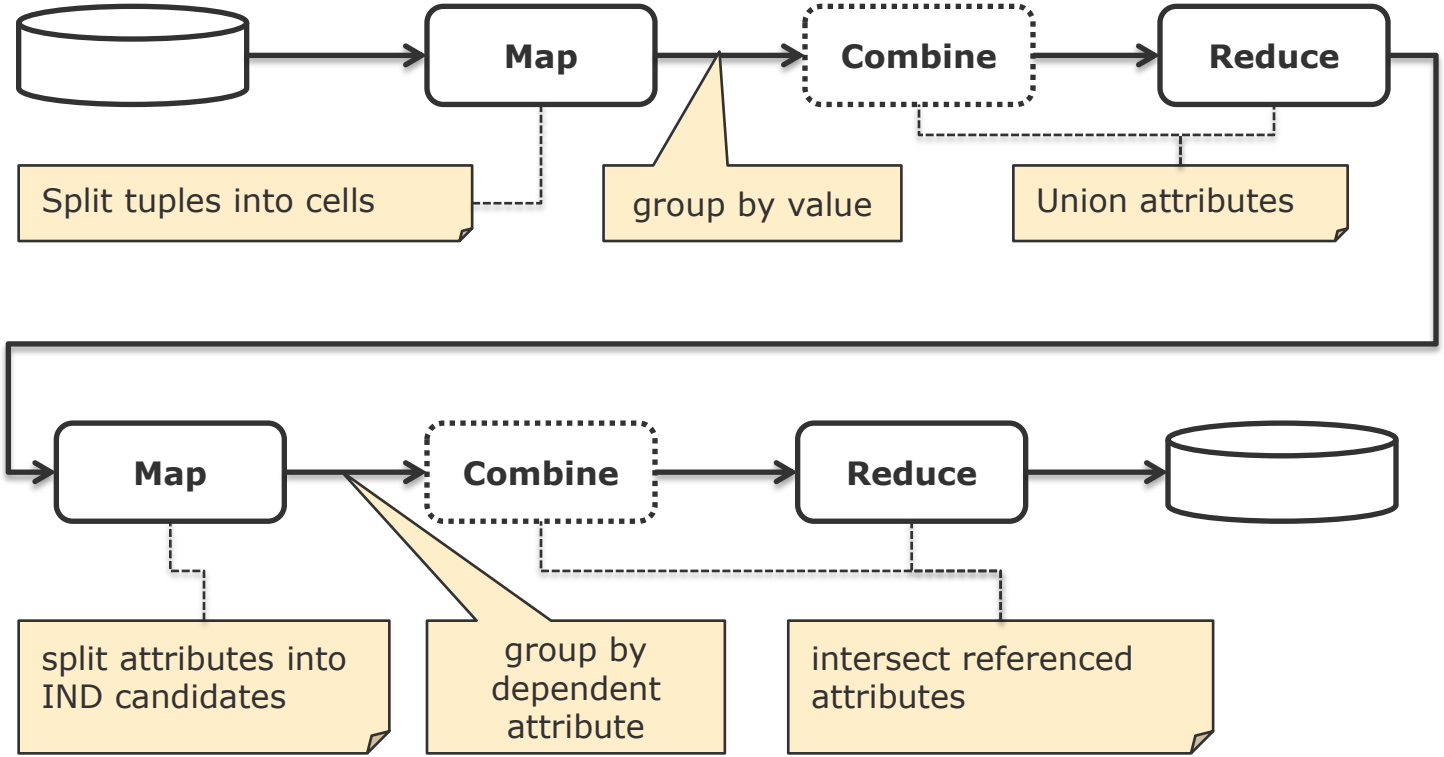
Backup

Apache Hadoop Implementation



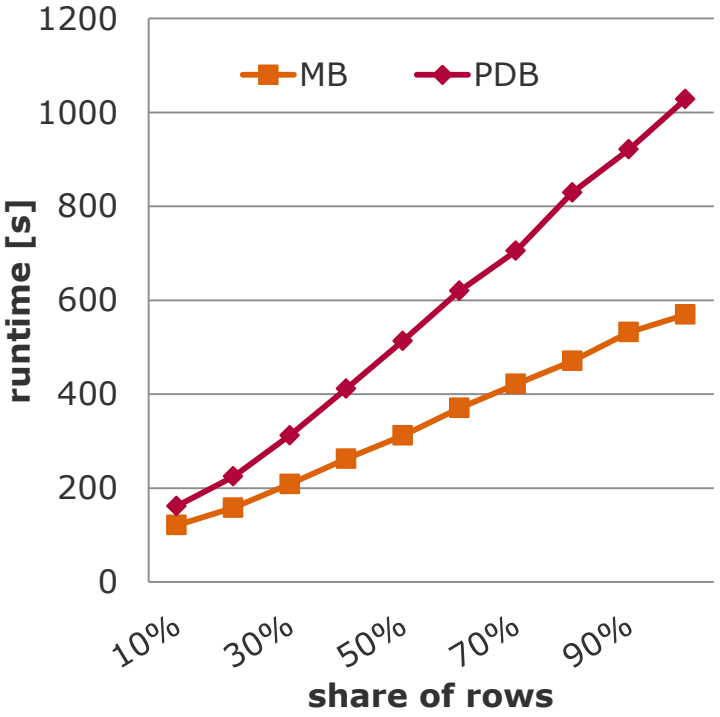
Backup

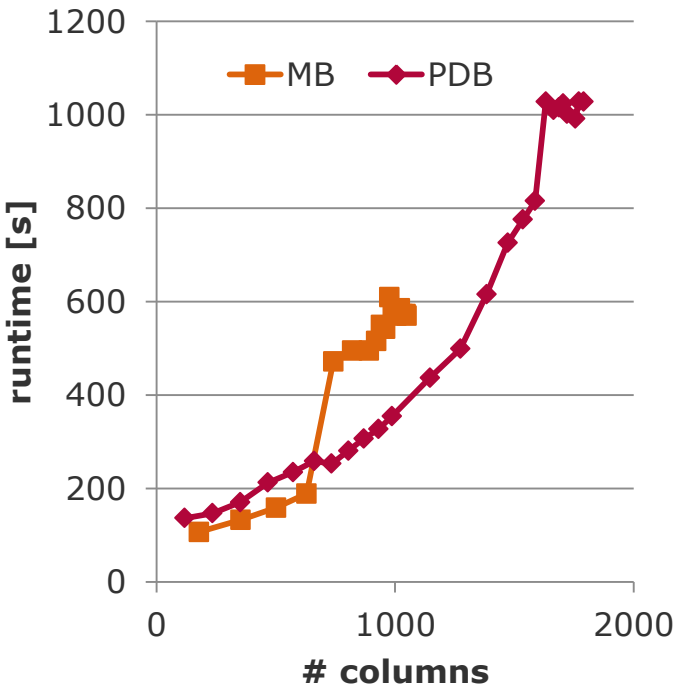
Apache Flink/Spark Implementation



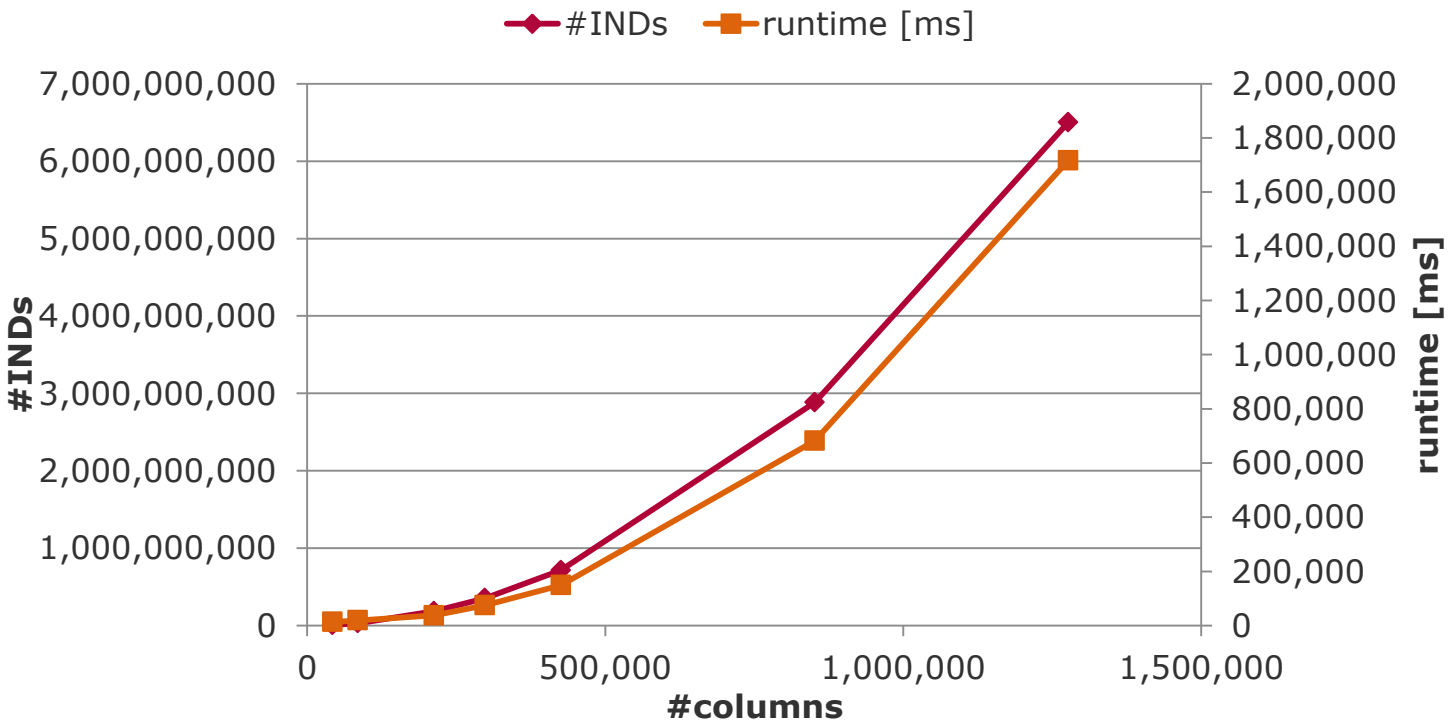
Backup

Column and row scaling behavior





Evaluation: Wikitables



Evaluation: n-ary INDs