

Query driven Entity Resolution in Data Lakes

Giorgos Alexiou, George Papastefanatos

Giorgos Alexiou
Information Management Systems Institute
Research Center “Athena”

galexiou@imis.athena-innovation.gr

Motivation


- A data warehouse is a repository for structured, filtered data that has already been processed for a specific purpose.
- A data lake is a vast pool of raw data that need to be processed to extract useful insights.
- Data analysis on voluminous data is a very expensive and slow task.
- Streaming Data

Motivation cont'd


- More than 80% of data scientists spend >50% of their time on cleaning and preparation of data
- Small organizations with large data sets and limited computational resources
- Data lakes usually contain duplicate entities from multiple heterogeneous sources which need to be resolved before enabling further analysis.

Query-Driven E.R.

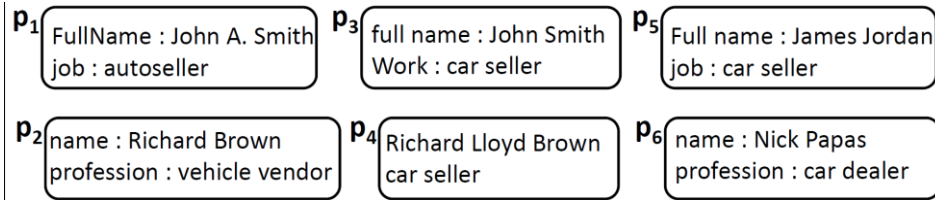
- **Traditional ETL processes:**

- Extract data from static data sources
- Transform and clean (Traditional ER methods)  Very expensive on voluminous data collections
- Load to DB
- Queries are on DB

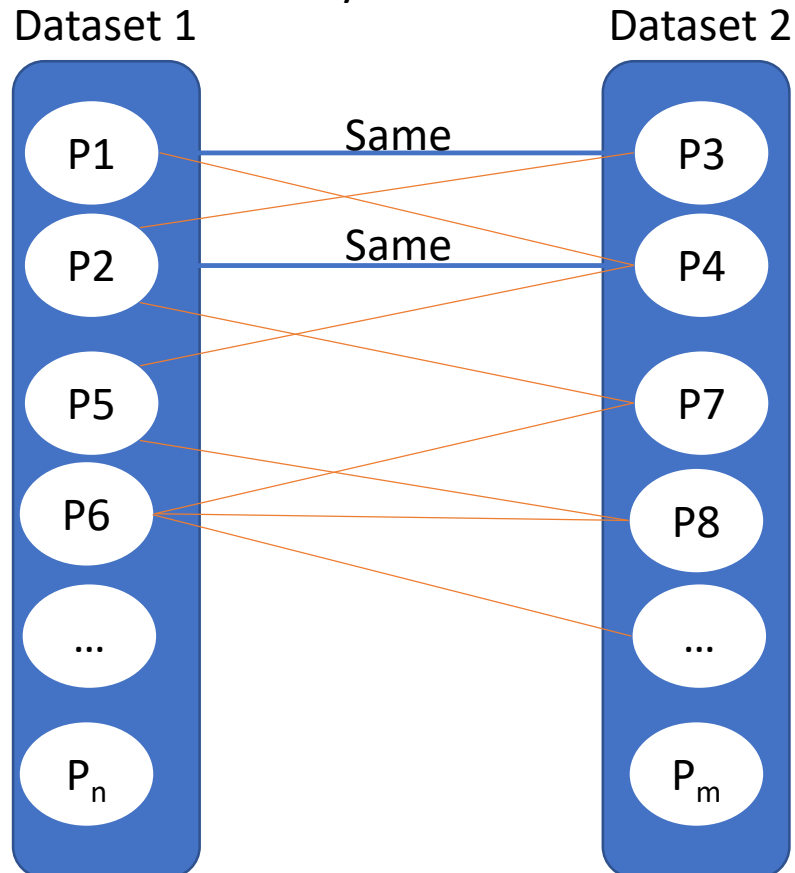
- **Query-Driven Entity Resolution**

- Seamlessly integrates query with raw data
- Extract clean data based on query  Useful especially in Data Lake environments
- Perform the necessary actions (e.g. analytics)

Entity Resolution (ER)



Entity Profiles



- Identifies and aggregates the **different** entity profiles that actually describe the **same** real-world object.
- Applications:
 - Duplicate detection – Dirty ER
 - Record linkage – Clean Clean ER

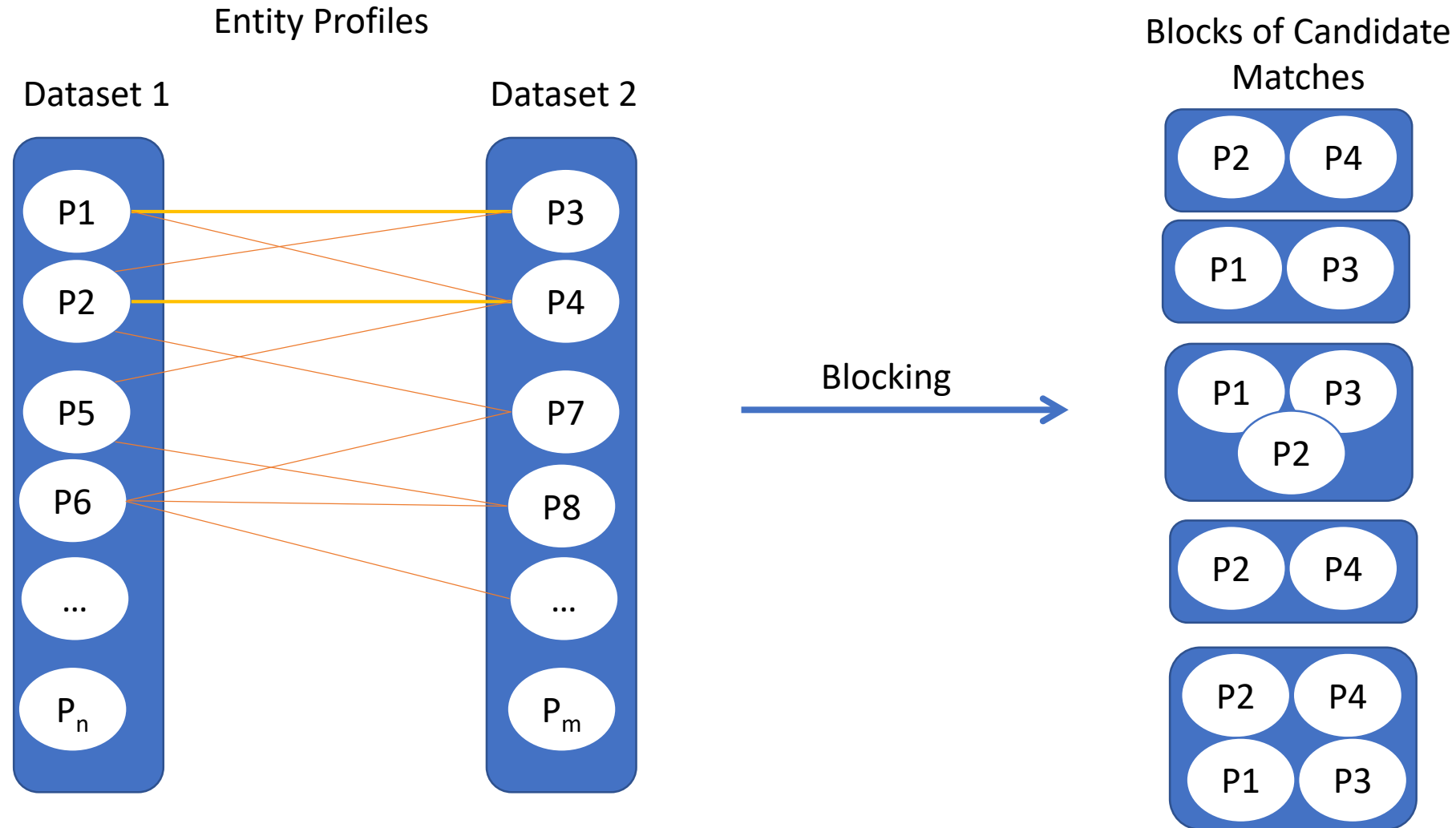
Time Complexity → quadratic

$$O(n^2)$$

Every entity is compared with all others.

Blocking

Most brute-force comparisons involve non-matching entities



Running Example

SI Index

id	Title	Author	Venue	Year
P1	Towards efficient Entity Resolution	Perry Scope	VLDB	2003
P2	Towards efficient E.R.	Perry Scope	Very Large Data Bases	
P3	Entity Resolution on web data	Allie Grater, John Doe	ACM SIGMOD	2016
P4	E.R. on web data	A. Grater, J. Doe	SIGMOD Conf	2016
P5	Entity-resolution on web data	A. Grater, John D.	Proc of ACM SIGMOD	
P6	Entity-Resolution for scholarly data	Perry Scope, Emma Grate	VLDB	2015
P7	E.R. for scholarly data	P. Scope, E. Grate	Very Large Data Bases	2015

- Select * FROM SI where venue =“VLDB”

Proposed Approach



ER-enriched Query-Plan

1. Select from the pre-computed similarity index that lies in the Data Lake
2. Build the blocking index based on the similarity of the entities' attributes employing schema-agnostic techniques
3. Perform Meta-blocking for reducing the final number of blocks, and thus the comparisons to be performed
4. Perform the actual matching on the remaining comparisons in blocks to resolve the duplicates
5. Merge the resolved entities into representative ones
6. The results can be further used by any subsequent operation (project, join, etc.) needed to answer the query

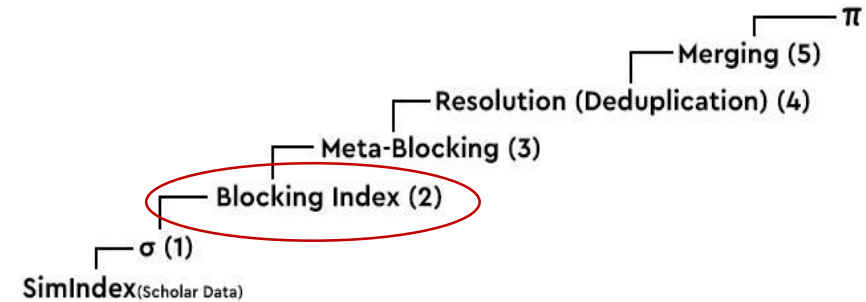
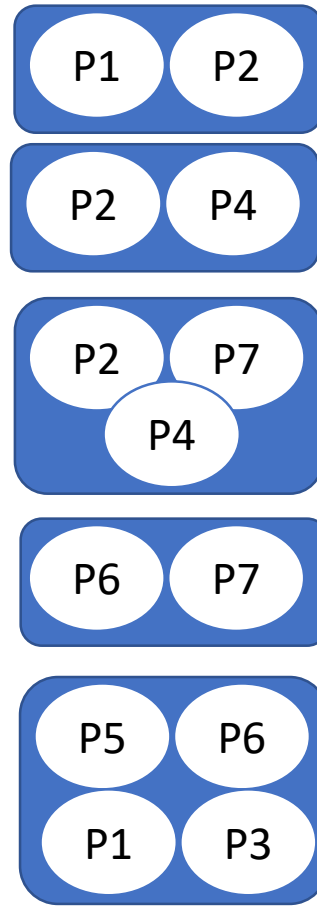
Blocking Index (2)

blocking Index

BK	Entities
to	P1,P2
wa	P1,P2
en	P1,P3,P5,P6
er	P2,P4,P7
	.
	.
sc	P6,P7
we	P3,P4,P5



Blocks of Candidate Matches



Goal:

Grouping entities into blocks based on the similarity of their attributes (e.g., prefixes or n-grams, tokens, etc.)

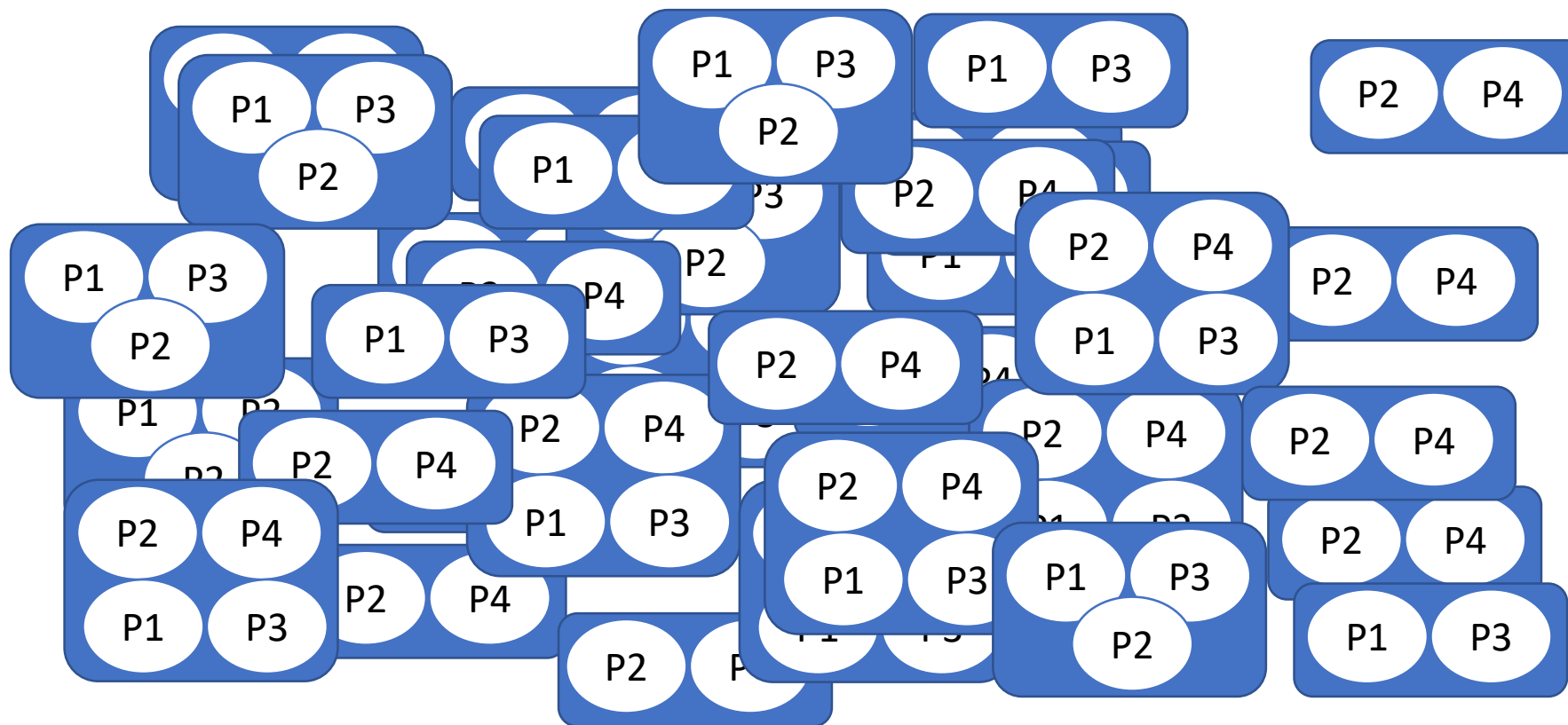
- We employ schema-agnostic blocking techniques, which are preferred for their increased effectiveness in terms of recall. [Papadakis et. al., VLDB 2015]

+ Schema-agnostic Blocking we can ensure that we won't miss any results

■ It creates many redundant blocks

Blocking in Data Lakes

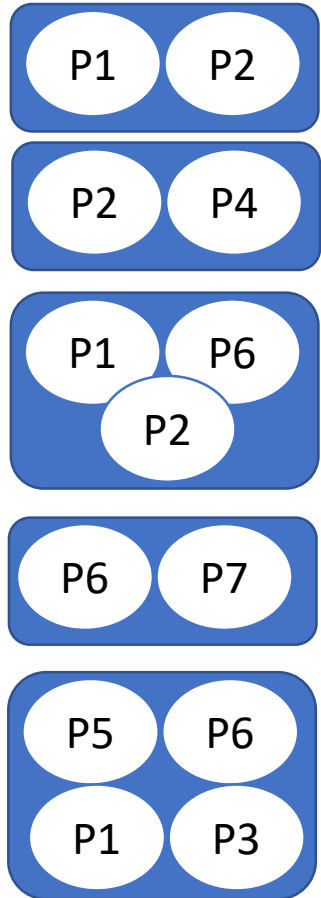
Too many blocks – too many comparisons



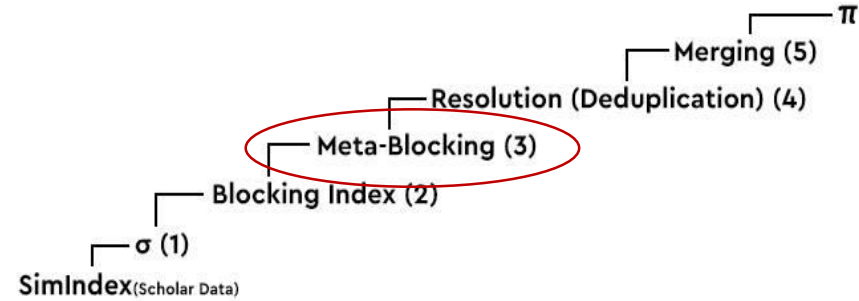
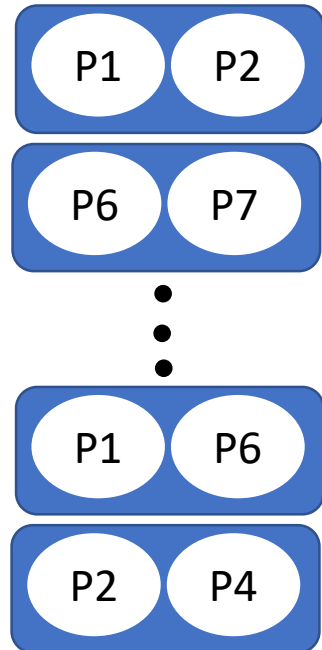
Prune blocks & comparisons without missing matching entities

Meta-Blocking (3)

Blocks of Candidate Matches



Blocks with fewer comparisons



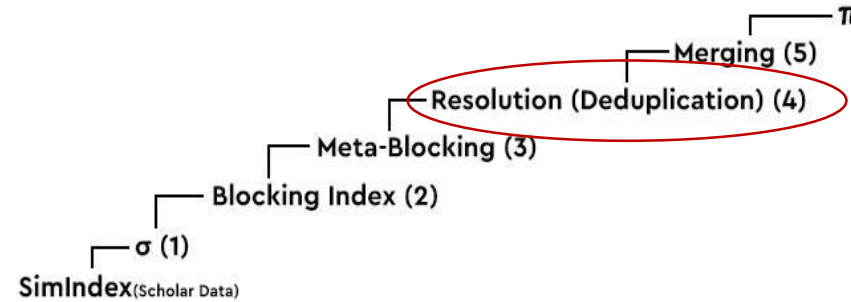
Every **comparison** between entity profiles p_i and p_j is:

1. **Matching** if $p_i \equiv p_j$.
2. **Redundant** if p_i and p_j co-occur and are compared in another block.
3. **Superfluous** if $p_i \neq p_j$ and the comparison is not redundant.

- **Goal of Meta-blocking** [Papadakis et. al., TKDE 2014]

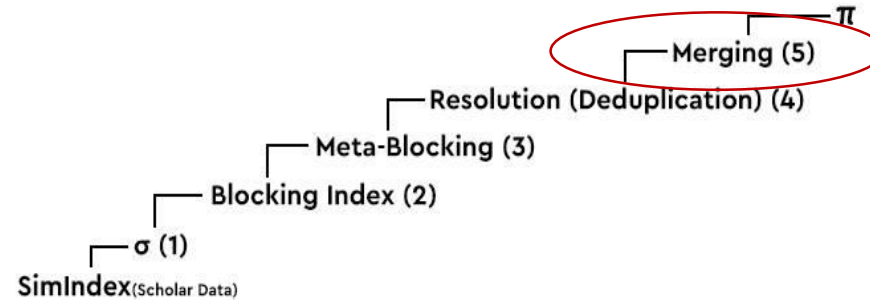
Given a given block collection B , restructure it into a new one that contains significantly fewer **redundant** and **superfluous** comparisons, while maintaining the original number of **matching** ones.

Resolution (4)



- The final comparisons (e.g. Jaccard, edit distance etc.) of the entities are performed on the remaining blocks to resolve the duplicates
- This phase determines for each pair of entities within the same block whether they co-refer or not.
- The resolved entities are clustered into non overlapping blocks to be used as input in step 5

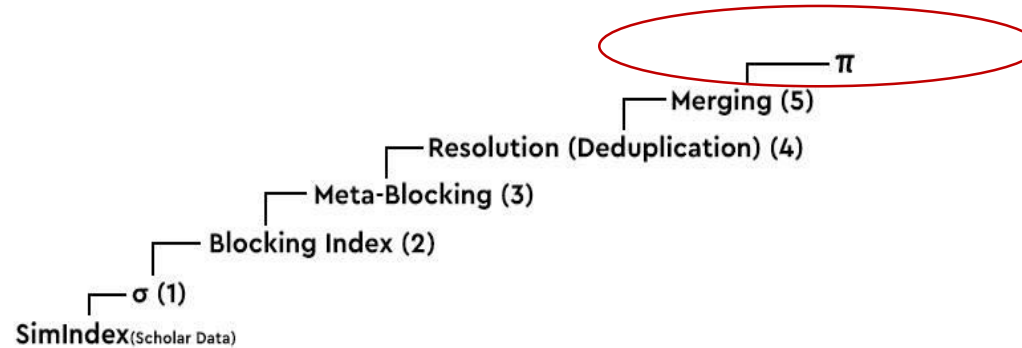
Merging (5)



- The resolved entities are merged into representative ones
- Merging phase combines entities of the resolved blocks into a single object that will represent the block to the end user in the final result
- A merge function will fuse the elements of a block B_i to produce a fused object O_i
- To merge the duplicate entities in blocks we will use functions based on entities' attributes that take multiple values and produce a single one.

id	Title	Author	Venue	Year
P1 ⊕ P2	Towards efficient Entity Resolution	Perry Scope	Very Large Data Bases	2003
P6 ⊕ P7	Entity-Resolution for scholarly data	Perry Scope, Emma Grate	Very Large Data Bases	2015

Output



- After this step, the results can be further used by any subsequent operation (project, join, analytics, etc.) needed to answer the query.
- The output (i.e. merged entities) can also be used to update the initial similarity index with the resolved ones to allow us to speed-up next queries by having a “cleaner” initial set.

Future Work

This is an ongoing work and we have not yet performed experiments on our methods.

Future work includes:

- Analytical experimental evaluation
- Automate the creation of the similarity index
- Scalability testing
- Parallel approach

Summary

we propose a method to:

- Seamlessly integrate Entity Resolution in query processing in the context of Data Lakes.
- Integrate schema-agnostic blocking methods in query processing
- Integrate Meta-blocking techniques in query processing

Such operations will enable users to perform more complex analytics on voluminous data sources avoiding the tedious tasks of data cleansing and integration.

Thank You!

Questions?