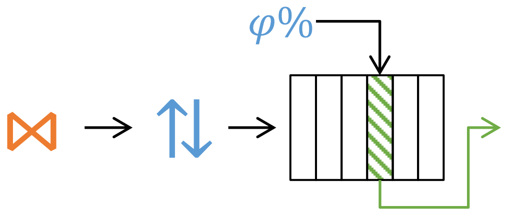
1 DATALAB
Northeastern University
Khoury College of Computer Sciences

2 Inria LIRMM
UNIVERSITÉ DE MONTPELLIER
cnrs

3 TECHNION | The Henry and Marilyn Taub Faculty of Computer Science

Efficient Computation of Quantiles over Joins

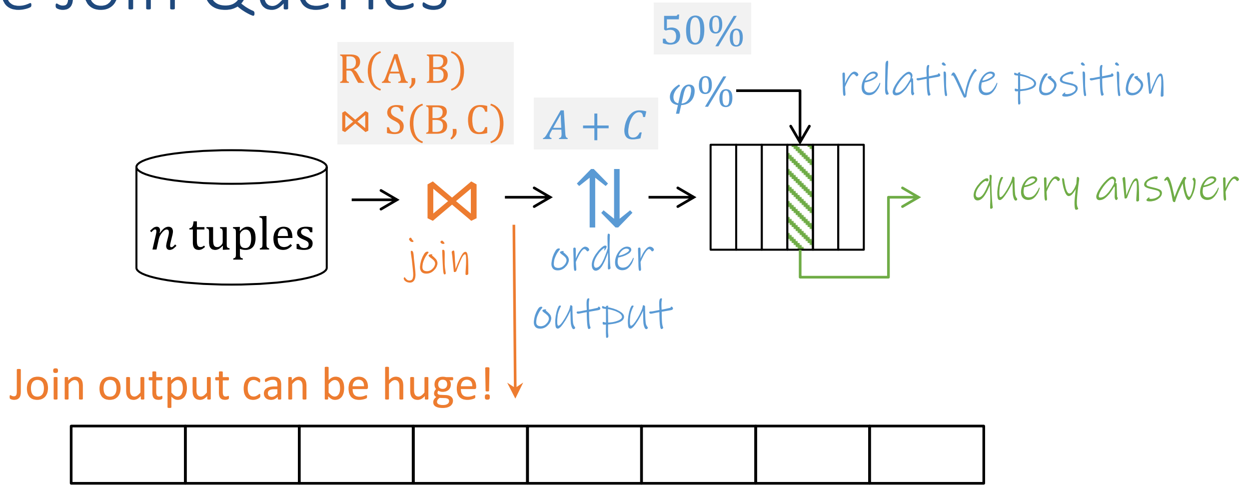
PODS 2023



Nikolaos Tziavelis[1],
Nofar Carmeli[2], Wolfgang Gatterbauer[1], Benny Kimelfeld[3], Mirek Riedewald[1]

https://northeastern-datalab.github.io/anyk/

Quantile Join Queries



$\mathrm{R(A,B)} \bowtie \mathrm{S(B,C)} \longrightarrow \mathrm{O}(\mathrm{n}^2)$

$\mathrm{R(A,B)} \bowtie \mathrm{S(A,C)} \bowtie \mathrm{T(A,D)} \longrightarrow \mathrm{O}(\mathrm{n}^3)$

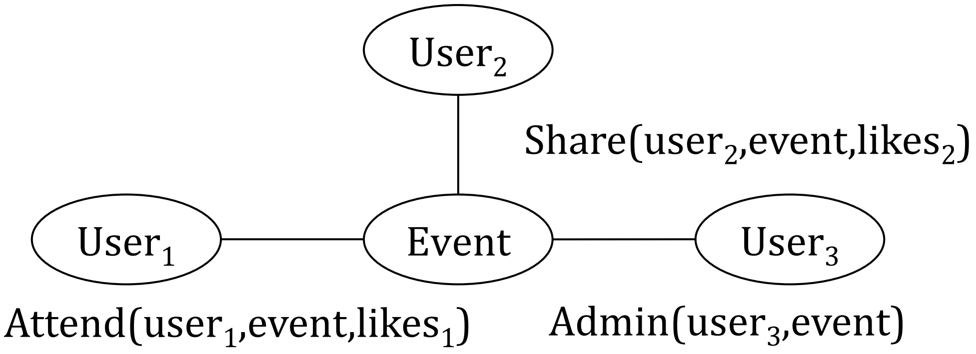
…

$\mathrm{O}(\mathrm{n}^d)$ for some constant $d \geq 1$, determined by the AGM bound of the query [A+08]

Main question: When can we find the quantile without computing the join?

[A+08] Atserias, Grohe, Marx. Size bounds and query plans for relational joins. *FOCS'08* https://doi.org/10.1109/FOCS.2008.43

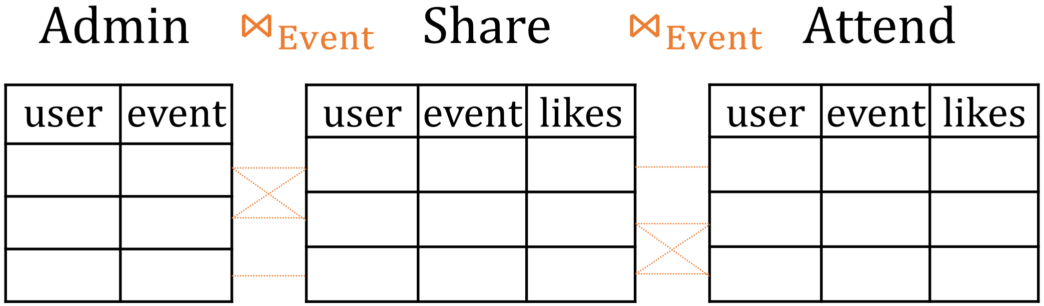
Example: Event Social Network Query

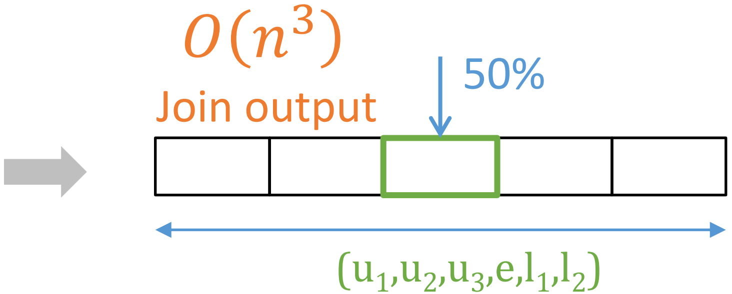


<u>Statistics</u> for this pattern?

1) Join Query
2) Sort by:
- likes$_1$ + likes$_2$
- MAX(likes$_1$, likes$_2$)
- ...

3) Select median query answer





We show that it can be done in $O(n \text{ polylog } n)$ without computing the join whose size is $O(n^3)$

Quantile Join Query Problem

Join query

$R(A, B), S(B, C), T(C, D)$
$\sigma_\theta(R \bowtie S \bowtie T)$

```
select    R.A, R.B, S.C, T.D,
          R.A+R.B+S.C+T.D as Σw
from      R, S, T
where     R.B=S.B and S.C=T.C
order by  Σw ASC
```

Ranking function

- SUM, MIN, MAX over weighted attributes
- (LEX)icographic orders of attributes

%JQ problem
- Input: database D of size n, relative position $\varphi \in [0,1]$
- Output: query answer at position $\lfloor \varphi |Q(D)| \rfloor$ in sorted array

Goal: achieve $O(n \text{ polylog } n)$ data complexity

- even though join output size is $O(n^d)$

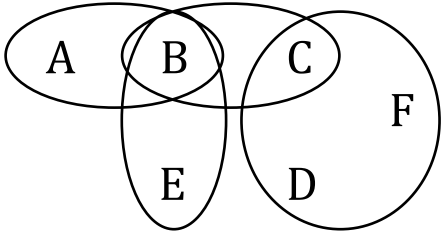
Outline

- Motivation & Problem Definition
- Prior Work
- New Results
- Algorithmic Framework
- Conclusion

Basic Definitions

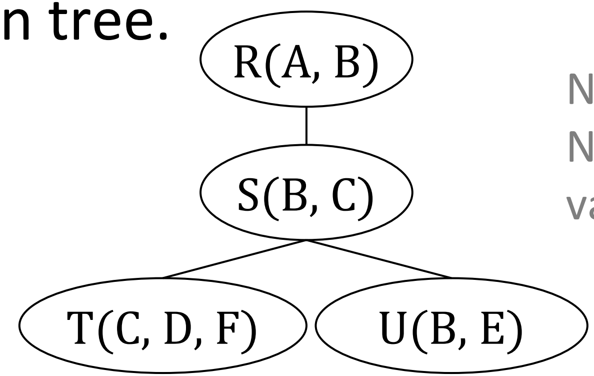
1. A JQ can be represented by a hypergraph.

$R(A,B), S(B,C), T(C,D,F), U(B,E)$



Nodes=Variables
Hyperedges=Atoms

2. A JQ is acyclic if it admits a join tree.

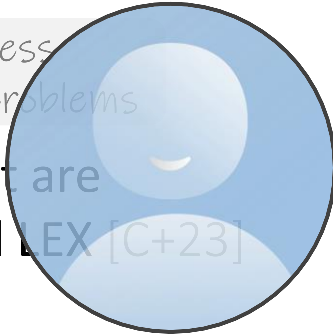


Nodes= Atoms
Nodes containing same variable are connected

3. A JQ is self-join-free if no relation appears twice.

Prior Dichotomy Results

Conditional on hardness hypotheses for certain problems

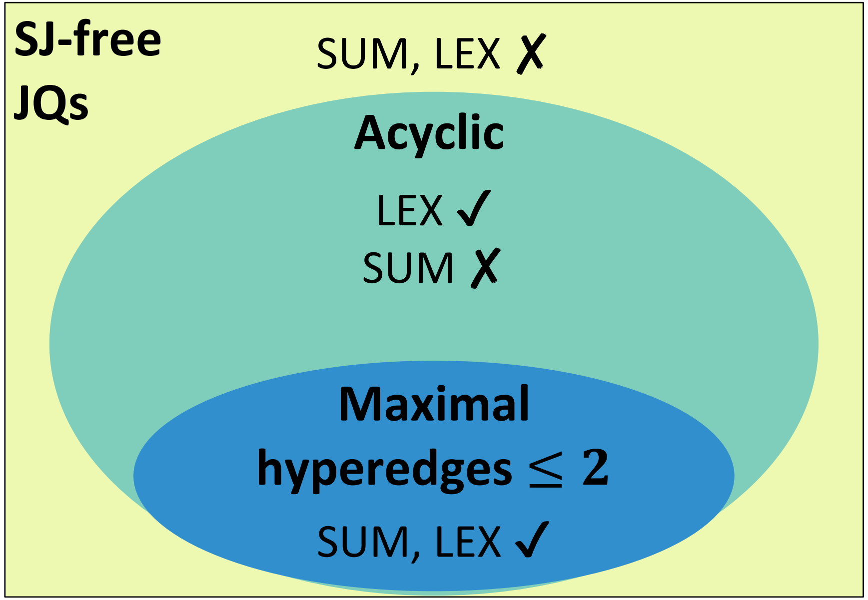


- Our prior work characterized precisely the (self-join-free) queries that are tractable (i.e., $O(n \operatorname{polylog} n)$ time) for 2 ranking functions: SUM and LEX [C+23]

$\mathrm{R(A, B), S(B, C), T(C, A)}$

$\mathrm{R(A, B), S(B, C), T(C, D)}$

$\mathrm{R(A, B), S(B, C)}$



<u>The end of the story?</u>

☹ Limited tractable cases for SUM

☹ Specialized algorithms for LEX/SUM

☹ Not clear how to apply to other ranking functions

[C+23] Carmeli, Tziavelis, Gatterbauer, Kimelfeld, Riedewald. Tractable Orders for Direct Access to Ranked Answers of Conjunctive Queries. *PODS'21*, *TODS'23*
https://doi.org/10.1145/3578517

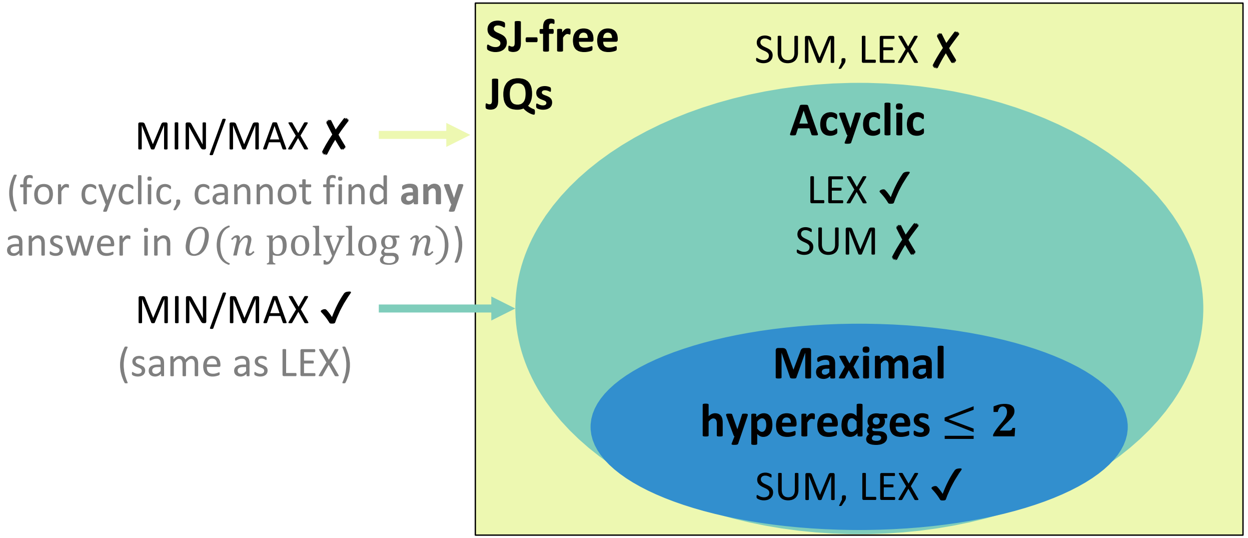
Outline

- Motivation & Problem Definition
- Prior Work
- New Results
- Algorithmic Framework
- Conclusion

New Results: 1) MIN/MAX

- We develop a general algorithmic framework that applies to all ranking functions mentioned (SUM, MIN, MAX, LEX). We use it to establish all our new results.

Theorem 1: %JQ with MIN/MAX is tractable for all acyclic queries.



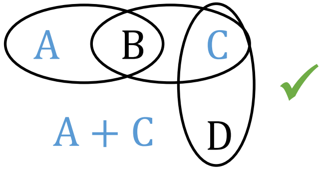
+ Same framework recovers old results up to a log factor

New Results: 2) Partial SUM

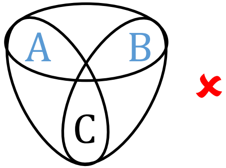
- Prior dichotomy assumed the worst-case SUM for each query where all attributes (variables) participate in ranking.
- We refine the SUM dichotomy by considering queries with partial SUMs.
  + Positive: We apply our framework. Prior algorithm specific to 2 relations only.
  - Negative: We prove conditional lower bounds.

Theorem 2: %JQ for self-join-free queries with partial SUM is tractable if and only if:

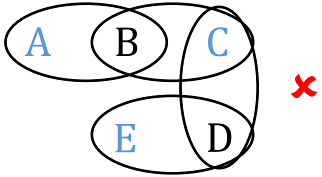
1. The query is acyclic.
2. There are at most 2 independent SUM variables.
3. Any chordless path between SUM variables is of length at most 3.



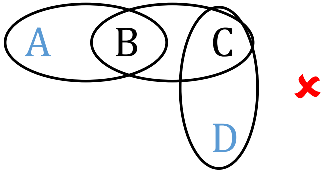
3 maximal hyperedges → intractable by prior dichotomy



A + B
cyclic



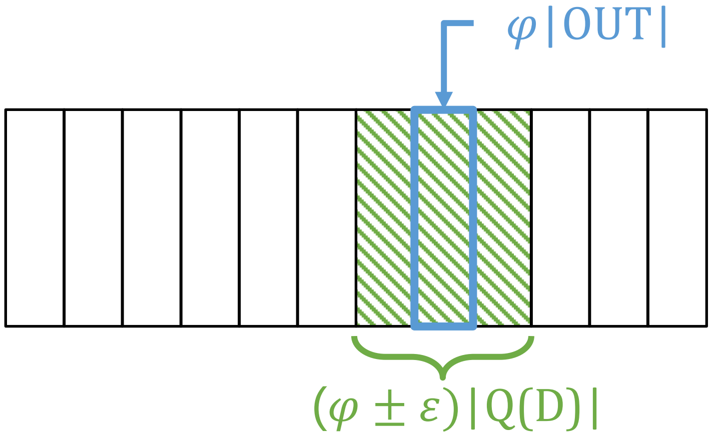
A + C + E
3 independent variables



A + D
Chordless path of length 4

New Results: 3) Approximate Quantiles for SUM

- ε-approximate quantiles: Given $\varepsilon \in (0,1)$, return $(\varphi \pm \varepsilon)$-quantile



$(0.5 \pm 0.01)|\mathrm{Q(D)}|$

$[49\% - 51\%]$

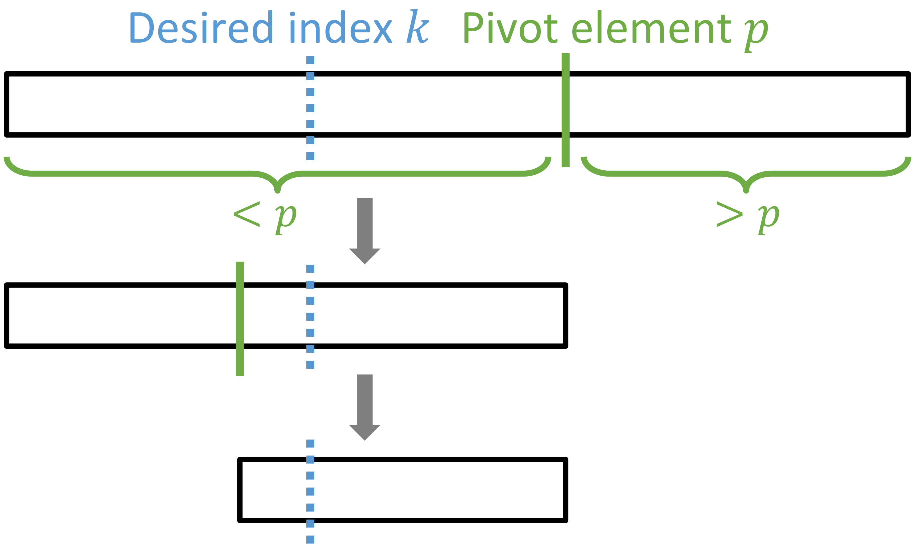
Same as LEX/MIN/MAX

Theorem 3: ε-approximate %JQ with (full or partial) SUM is tractable for all acyclic queries.

Outline

- Motivation & Problem Definition
- Prior Work
- New Results
- Algorithmic Framework
- Conclusion

Linear-Time Selection on an Array



Compare counts with k to decide which partition to keep

… until "few" elements left

Differences with our problem

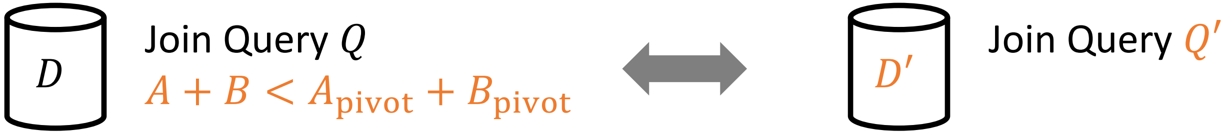
1. We do not have access to the array of query answers!
2. $O(n \log n) \rightarrow O(n)$ vs $O(n^d) \rightarrow O(n \operatorname{polylog} n)$
3. We can use linear-time selection as a subroutine.

[B+73] Blum, Floyd, Pratt, Tarjan. Time bounds for selection. *JCSS* 1973 https://doi.org/10.1016/S0022-0000(73)80033-9

Applying the Idea to %JQs

What do we need to apply the pivot-and-partition idea to %JQs?

1. Select pivot
   - A pivot is one of the query answers.
   - It needs to eliminate a constant fraction of remaining answers (to get convergence in logarithmic rounds)
2. Partition the query answers
   - We only have access to the database, not the answers!
   - Can be achieved by “trimming” inequalities



3. Count the answers in the $<$ and $>$ splits ✓
   - can be done in linear time for acyclic JQs

%JQ Framework

PIVOT

$O(n)$ algorithm for
“subset-monotone”
ranking functions
(SUM, MIN, MAX, LEX, …)

TRIM

Customized construction
for each ranking function

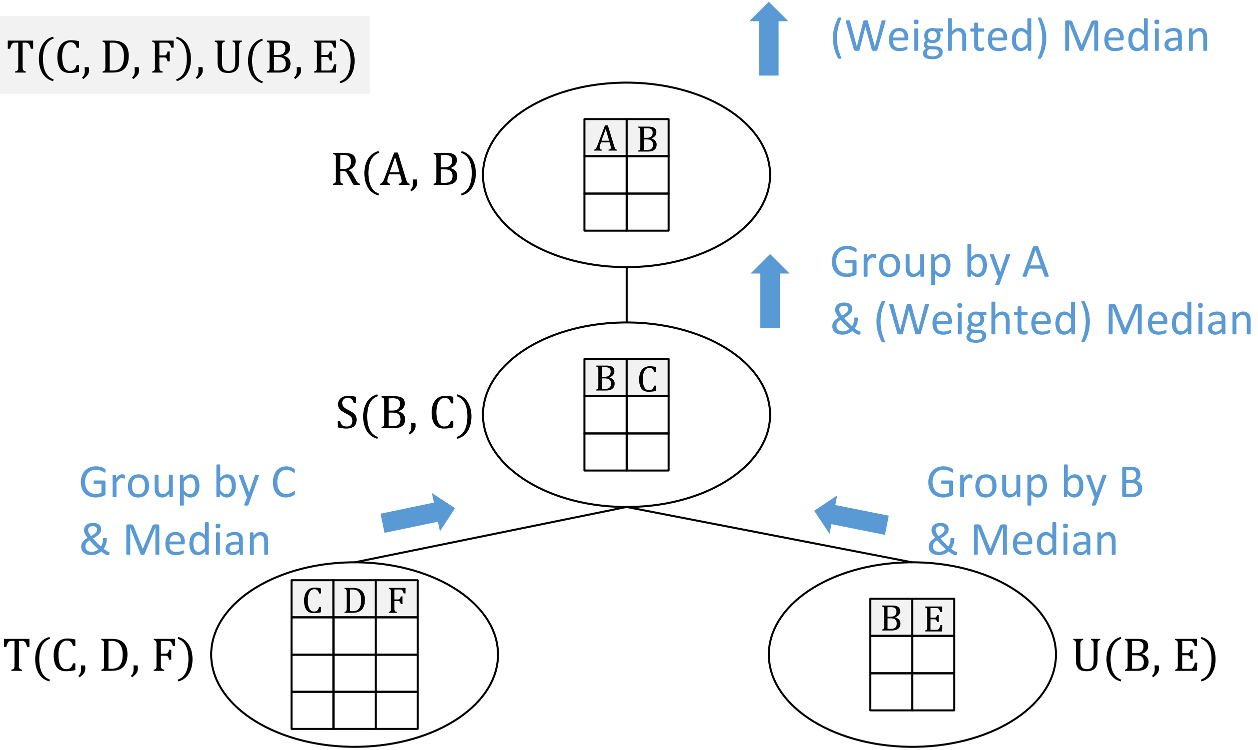
Lossy trimming → Approximate Quantiles!

Also eliminates some of the answers that do satisfy the inequality

Pivot Selection Algorithm

Message passing, bottom-up in the join tree.
Take (weighted) median at each level.

$R(A, B), S(B, C), T(C, D, F), U(B, E)$

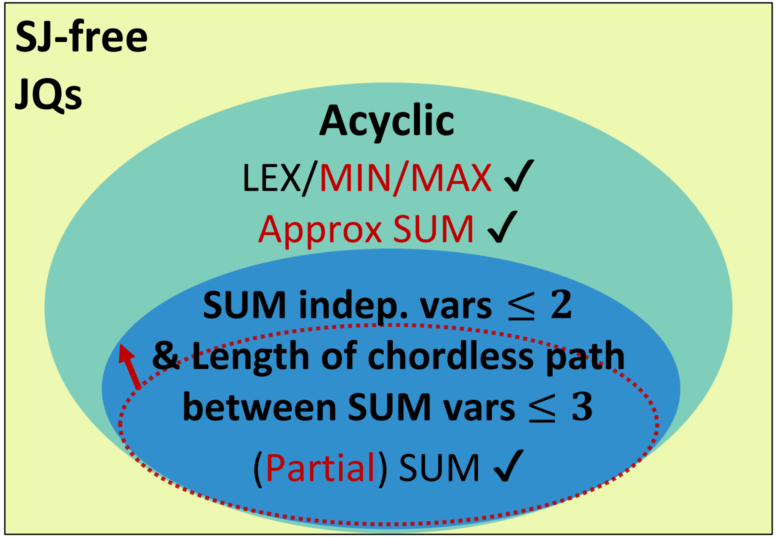


Outline

- Motivation & Problem Definition
- Prior Work
- New Results
- Algorithmic Framework
- Conclusion

Conclusion

- General framework for %JQs that reduces the problem of %JQ to that of trimming inequalities (for appropriately monotone ranking functions).
- Many cases where quantiles can be found in $O(n \text{ polylog } n)$ without materializing the join output.
  - Existing database systems may struggle with computing expensive joins.
- Our algorithms also apply to Conjunctive Queries (i.e., JQs with projections) as long as they are "free-connex".
  - Lower bounds for CQs are not 100% clear.



Thank you!