15

Bounded Approximation Scheme

Input: A resource ratio $\alpha \in (0, 1]$ and an access schema $\mathcal{A}$.

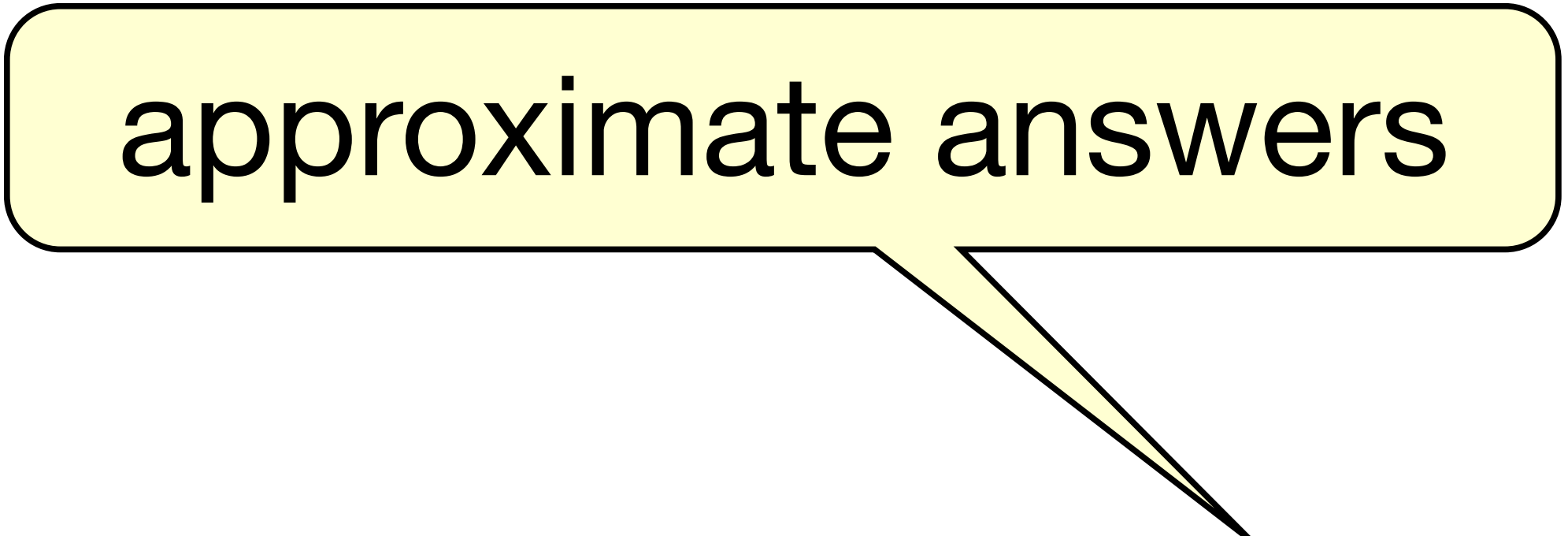
Scheme[6]: Given generic SQL Q and D, computes $(Q(D_Q), \eta)$:

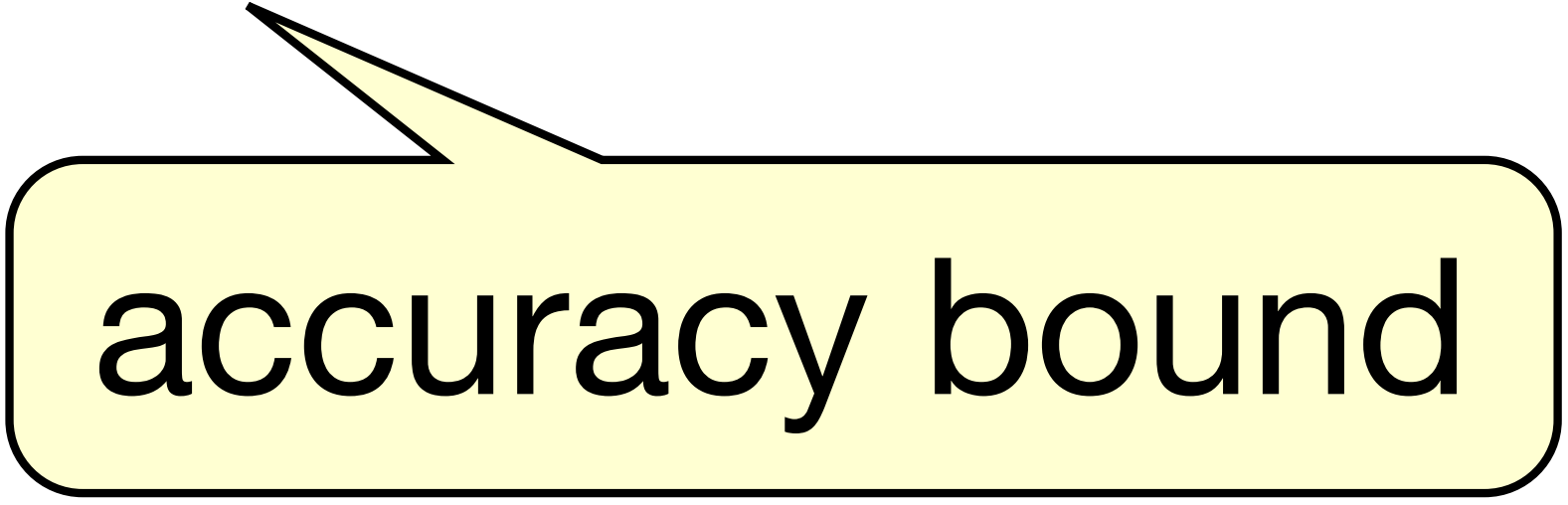
- access a fraction D_Q of D with $|\boldsymbol{D_Q}| \leq \boldsymbol{\alpha}|\boldsymbol{D}|$;
- $\textbf{accuracy}(\boldsymbol{Q}, \boldsymbol{D}, \boldsymbol{Q}(\boldsymbol{D_Q})) \geq \boldsymbol{\eta}$.

Chellenges:

- **Deterministic** bound for **generic** queries (even non-aggregate)
- Compute **both** answers and accuracy bound by accessing ≤ $\alpha|D|$ tuples

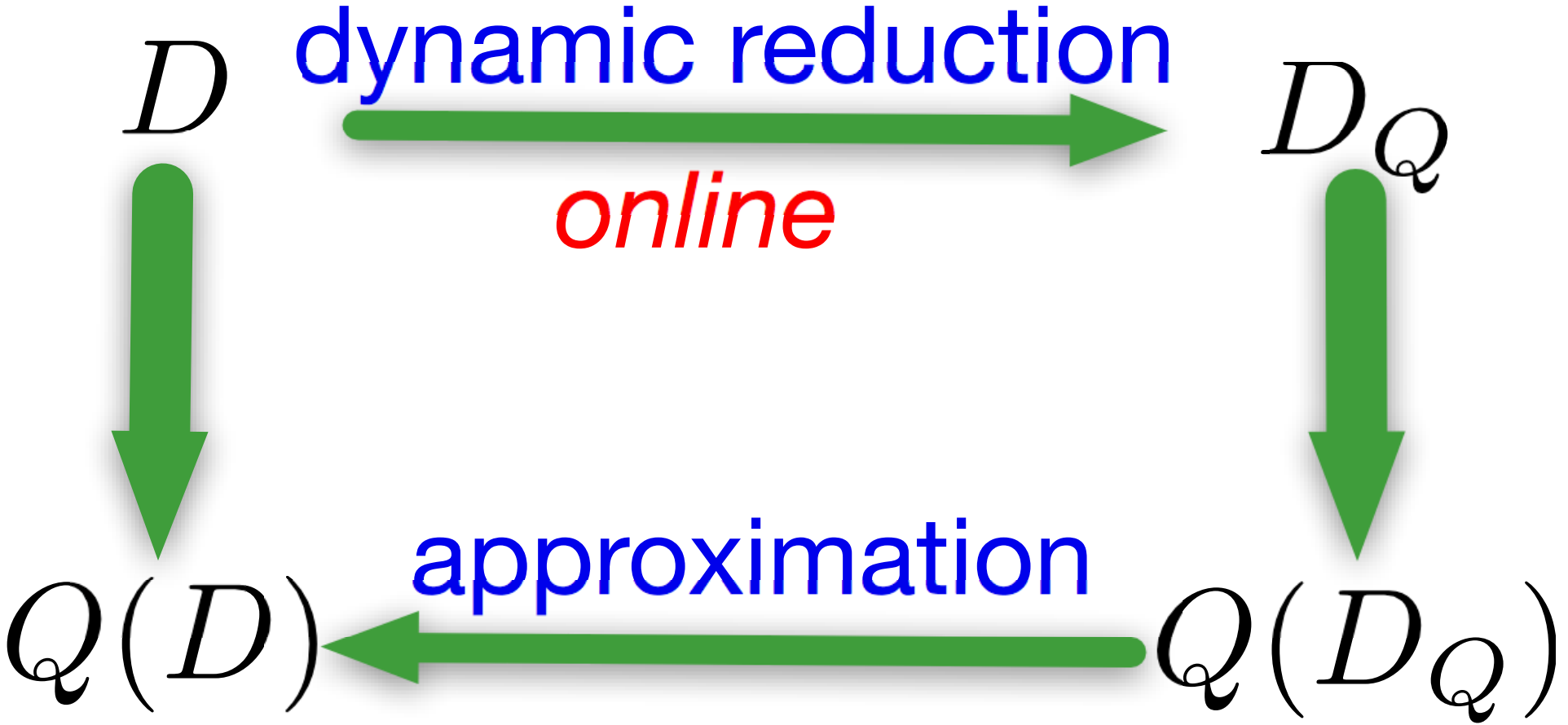
decided by available resources


approximate answers


accuracy bound

deterministic!

Flexible trade-offs: available resources vs. accuracy bound


D
dynamic reduction
D_Q
online
approximation
$Q(D)$
$Q(D_Q)$

[6] Y. Cao, W. Fan: Data Driven Approximation with Bounded Resources, VLDB 2017

not *one-size-fit-all* reduction

access data relevant to Q only to
make best use of available resources

Bounded Approximation Scheme

decided by available resources

approximate answers

Input: A resource ratio $\alpha \in (0, 1]$ and an access schema $\mathcal{A}$.

Scheme[6]: Given generic SQL Q and D, computes $(Q(D_Q), \eta)$:

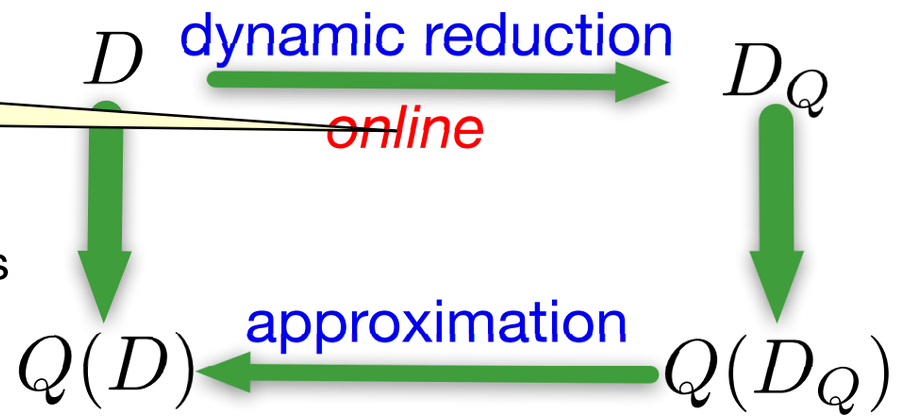
accuracy bound

- access a fraction D_Q of D with $|D_Q| \leq \alpha|D|$;
- $\mathbf{accuracy}(Q, D, Q(D_Q)) \geq \eta$.

deterministic!

not *one-size-fit-all* reduction

access data relevant to Q only to make best use of available resources



Flexible trade-offs: available resources vs. accuracy bound

Chellenges:

- **Deterministic** bound for **generic** queries (even non-aggregate)
- Compute **both** answers and accuracy bound by accessing ≤ $\alpha|D|$ tuples

[6] Y. Cao, W. Fan: Data Driven Approximation with Bounded Resources, VLDB 2017

How does it work?