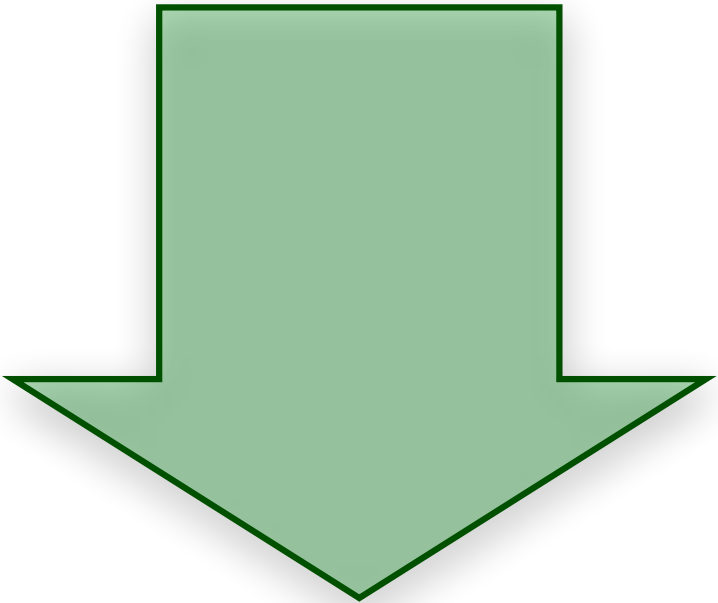
- **RC-measure**:
  - Relevance: every approximate answer in S is η-close to an exact answer in Q(D)
  - Coverage: every exact answer in Q(D) is η-close to an approximate answer in S

16

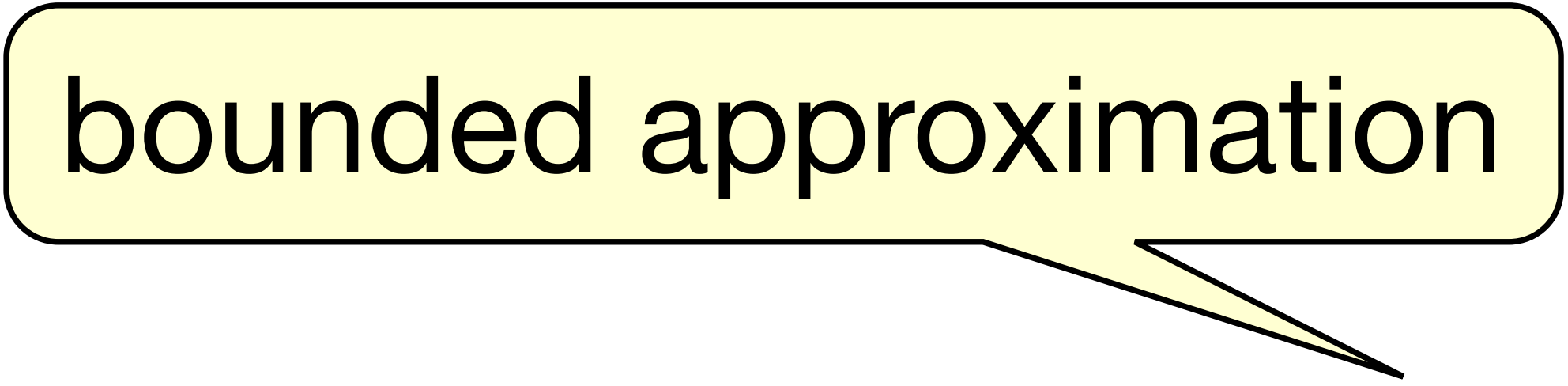
How does it work?

*Store "models" (***access constraints + data oracles***) instead of datasets*



*Store "approximate models" (***access templates + data oracles***)*

bounded evaluation


bounded approximation

- access templates: $\varphi = R(X \to Y, 2^k, \mathsf{acc}(k))$
- data oracles: fetch$(\bar{x}, k, \varphi)$ returns 2^k Y-values with accuracy $\mathtt{acc}(k)$

fetch *part of the* Y-values
with accuracy guarantee

$$\mathrm{acc}(S, Q(D)) \leq \eta$$

The first (only) accuracy measure for

- *resource-bounded* APQ / sub-linear plans
- generic queries (*e.g., set-valued* answers)

for *generic* queries

Features: (1) accessing ≤ α|D| tuples; (2) deterministic accuracy bound

How does it work?

bounded evaluation

*Store "models" (***access constraints + data oracles***) instead of datasets*

bounded approximation

*Store "approximate models" (***access templates + data oracles***)*

- access templates: $\varphi = R(X \to Y, 2^k, \mathrm{acc}(k))$

fetch *part of the* Y-values with accuracy guarantee

- data oracles: fetch$(\bar{x}, k, \varphi)$ returns 2^k Y-values with accuracy $\mathrm{acc}(k)$

- **RC-measure**: $\mathrm{acc}(S, Q(D)) \leq \eta$
  - Relevance: every approximate answer in S is η-close to an exact answer in Q(D)
  - Coverage: every exact answer in Q(D) is η-close to an approximate answer in S

The first (only) accuracy measure for
- *resource-bounded* APQ / sub-linear plans
- generic queries (e.g., set-valued answers)

for *generic* queries

Features: (1) accessing ≤ $\alpha|D|$ tuples; (2) deterministic accuracy bound

BEAS (Bounded EvAluation of Sql)

— Bounded Evaluation + Bounded Approximation