

Basic of Bundled Evaluation



Access constraint: $\phi \equiv R(X \rightarrow Y, N)$

► For each X -value, there are at most N distinct Y values in R

$$\phi_1 \equiv \text{friendd}(\text{uid} \rightarrow \text{fid}, 5000)$$

$\phi_2 \Rightarrow \text{update}(\text{uid} \rightarrow \text{country}, 193)$

Data oracle: *attribute-based* data access $\text{fetch}(\bar{x}, \phi)$

► given X -value, retrieve associated Y -values with bounded cost (N)

`fetch(uid1, ϕ_1)` returns all my friends, by
accessing ≤ 5000 tuples

```
SELECT fid  
FROM    friend AS f, update AS u  
WHERE   f.uid= uid1 AND f.fid=u.uid AND u.country=UK
```

$\text{fetch}(\text{uid2}, \phi_2)$ returns all countries for uid2,
by accessing ≤ 193 tuples

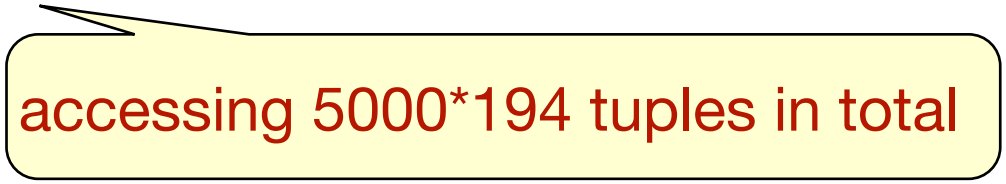
Store “models” (access constraints + data oracles) instead of datasets

from data semantics or aggregation of datasets

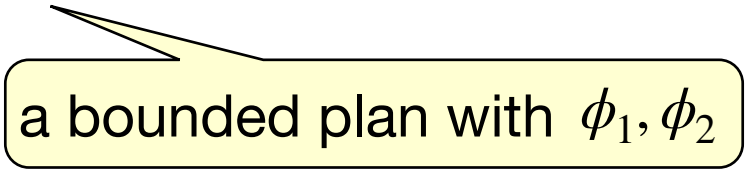
- `friend(uid, fid)`
- `update(id, uid, country, date, post, ...)`

boundedly evaluable with ϕ_1, ϕ_2

- $T_1 = \text{fetch}(\text{uid1}, \phi_1)$ fetch all my friends
(accessing at most 5000 friend tuples)
- $T_2 = \text{fetch}(T_1, \phi_2)$ all countries for my friends
(accessing at most 5000×193 update tuples)
- $T_3 = \pi_{\text{uid}} \sigma_{\text{country}=\text{UK}} T_2$



accessing 5000*194 tuples in total



a bounded plan with ϕ_1, ϕ_2

Boundedly evaluable queries can be answered within bounded cost

scale independence



Basics of Bounded Evaluation

Store “models” (access constraints + data oracles) instead of datasets

from data semantics or aggregation of datasets

Access constraint: $\phi = R(X \rightarrow Y, N)$

- ▶ For each X-value, there are at most N distinct Y values in R

Data oracle: attribute-based data access $\text{fetch}(\bar{x}, \phi)$

- ▶ given X-value, retrieve associated Y-values with bounded cost (N)

boundedly evaluable with ϕ_1, ϕ_2

- `friend(uid, fid)`
- `update(id, uid, country, date, post, ...)`

```
SELECT fid
FROM   friend AS f, update AS u
WHERE  f.uid= uid1 AND f.fid=u.uid AND u.country=UK
```

$\phi_1 = \text{friend}(\text{uid} \rightarrow \text{fid}, 5000)$

$\phi_2 = \text{update}(\text{uid} \rightarrow \text{country}, 193)$

$\text{fetch}(\text{uid1}, \phi_1)$ returns all my friends, by
accessing ≤ 5000 tuples

- $T_1 = \text{fetch}(\text{uid1}, \phi_1)$ fetch all my friends
(accessing at most 5000 friend tuples)

- $T_2 = \text{fetch}(T_1, \phi_2)$ all countries for my friends
(accessing at most 193 tuples)

scale independence

- $T_3 = \text{fetch}(T_2, \phi_3)$

Boundedly evaluable queries can be answered within **bounded** cost

Bounded Evaluability

- ▶ Input: A Query Q and an *access schema* A
- ▶ Question: Is Q boundedly evaluable with A