EDBT/ICDT 2020

Facets of Probabilistic Databases

Benny Kimelfeld

Technion Data & Knowledge Lab

tdk.cs.technion.ac.il

DFG Deutsche Forschungsgemeinschaft

TD&K TECHNION DATA & KNOWLEDGE LABORATORY

Technion Israel Institute of Technology

BSF United States – Israel Binational Science Foundation

ISRAEL SCIENCE FOUNDATION

Origins of Probabilistic Databases

- Traditional DB focused on **computational logic**
  - Worst-case guarantees, logical equivalence, …
- Other data fields adopted **statistics** foundations
  - IR, Web, NLP, ML, AI, …
- Natural questions arose:
  - *Push more of the analytics flow into DB?* (In-DB-X…)
  - *DB capabilities to the rescue of statistical problems?*

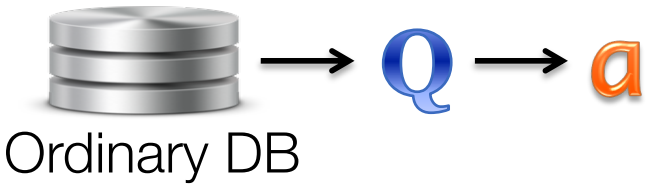
⇒ **Probabilistic Databases**

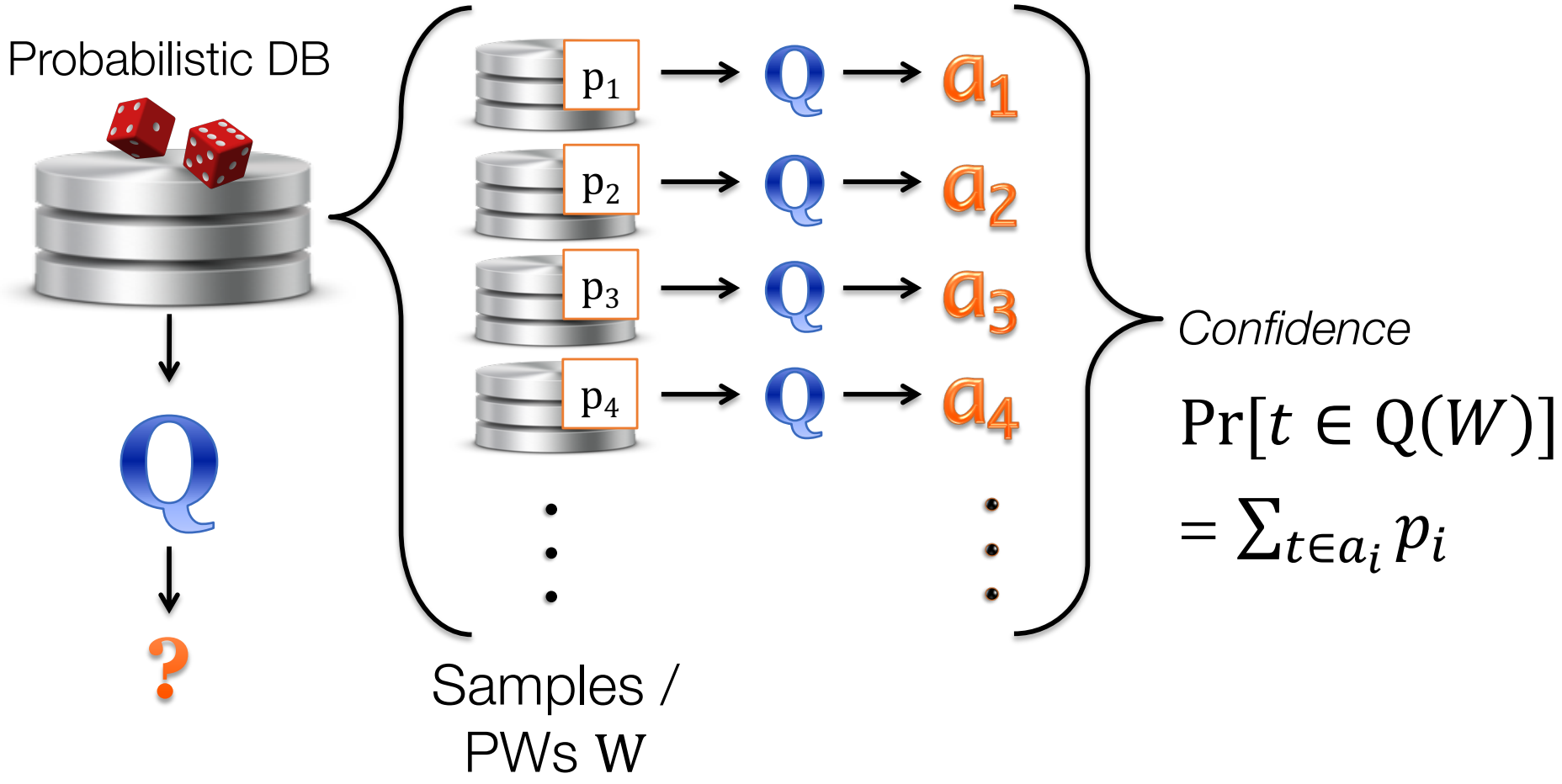
- Similar phenomenon in the PL community – push into compilers

⇒ **Probabilistic Programming**

- Not just a passing trend – nowadays studied from different angles in connection to various challenges

Query Answering as Marginal Inference





Example: Tuple-Independent DB (TID)

- Basic interpretation confidence-annotated tuples

person	city	state	p
Cullen	LA	CA	0.6
Cullen	Tampa	FL	0.4
Marion	LA	CA	1.0
Irene	NYC	NY	0.3
Irene	LA	FL	0.4

person	qualification	p
Cullen	9	0.3
Cullen	5	0.7
Marion	8	1.0
Irene	9	0.8

[deRougemont95] [Grädel-Gurevich-Hirsch98] [Dalvi-Suciu04]

Example: NELL

http://rtw.ml.cmu.edu/rtw/

Recently-Learned Facts

twitter

Refresh

instance	iteration	date learned	confidence
ramada_limited_airport_north is a tourist attraction	1111	06-jul-2018	94.3
pick_sports is a sport	1111	06-jul-2018	99.9
carlos_orellana is american	1111	06-jul-2018	91.6
harlan_friedman is a person	1111	06-jul-2018	100.0
chilled_sugar is a fruit	1111	06-jul-2018	99.7
the coach kurt_warner won the trophy or tournament super_bowl	1112	24-jul-2018	100.0
theodor_kolek is a person who has residence in the city jerusalem	1112	24-jul-2018	100.0
matt_moore plays in the league nfl	1116	12-sep-2018	94.7
chairs is often found in common_area	1111	06-jul-2018	96.9
marta_vincenzi is the leader of the city genoa	1116	12-sep-2018	100.0

Example: Tuple-Independent DB (TID)

- Basic interpretation confidence-annotated tuples
- Independence:

$$\Pr[W] = \prod_{t \in W} p(t) \times \prod_{t \notin W} (1 - p(t))$$

Subset of tuples

person	city	state	p
Cullen	LA	CA	0.6
Cullen	Tampa	FL	0.4
Marion	LA	CA	1.0
Irene	NYC	NY	0.3
Irene	LA	FL	0.4

person	qualification	p
Cullen	9	0.3
Cullen	5	0.7
Marion	8	1.0
Irene	9	0.8

[deRougemont95] [Grädel-Gurevich-Hirsch98] [Dalvi-Suciu04]

Practical Realizations

- *Implemented systems*
  - "Classic": **Trio** [Widom+], **ProbLog** [DeRaedt-Kimmig+], **MayBMS** [Koch+], **MystiQ** [Ré-Suciu+], **ProvSQL** [Senellart+]
  - Simulation systems: **MCDB**, **SimSQL** [Haas-Jermaine+]
  - Markov Logic / soft constraints: **Alchemy** [Domingos+], **SlimShot** [Gribkoff-Suciu], **DeepDive** [Ré+], **ForcLift** [Van den Broeck+], **ProbKB** [Chen+Wang], **Pr. Datalog+/-** [Gottlob+]
  - Data cleaning: **HoloClean** [Rekatsinas+]
- *Applications* text analytics, data cleaning, approx. query processing / subsampling [Zheng+14], statistical relational learning [VanHaaren+16], image retrieval [Zhu+15]
- *Public data* **Reverb** [Fader+11], **YAGO3** [Mahdisoltani+15], **NELL** [Carlson+15], MS **Concept Graph** [Wang+15]

Agenda

~~Build Better Systems~~

Probabilistic DBs as a guiding principle for DB theory

(personal perspective)

Plan

1. Inference over independent tuples
2. Broader perspective on PDB models
3. Reflections on database repairing

Tuple Independence

The complexity of fundamental DB problems is captured by inference over tuple-independent DBs

Similar Concepts in Logic Programming

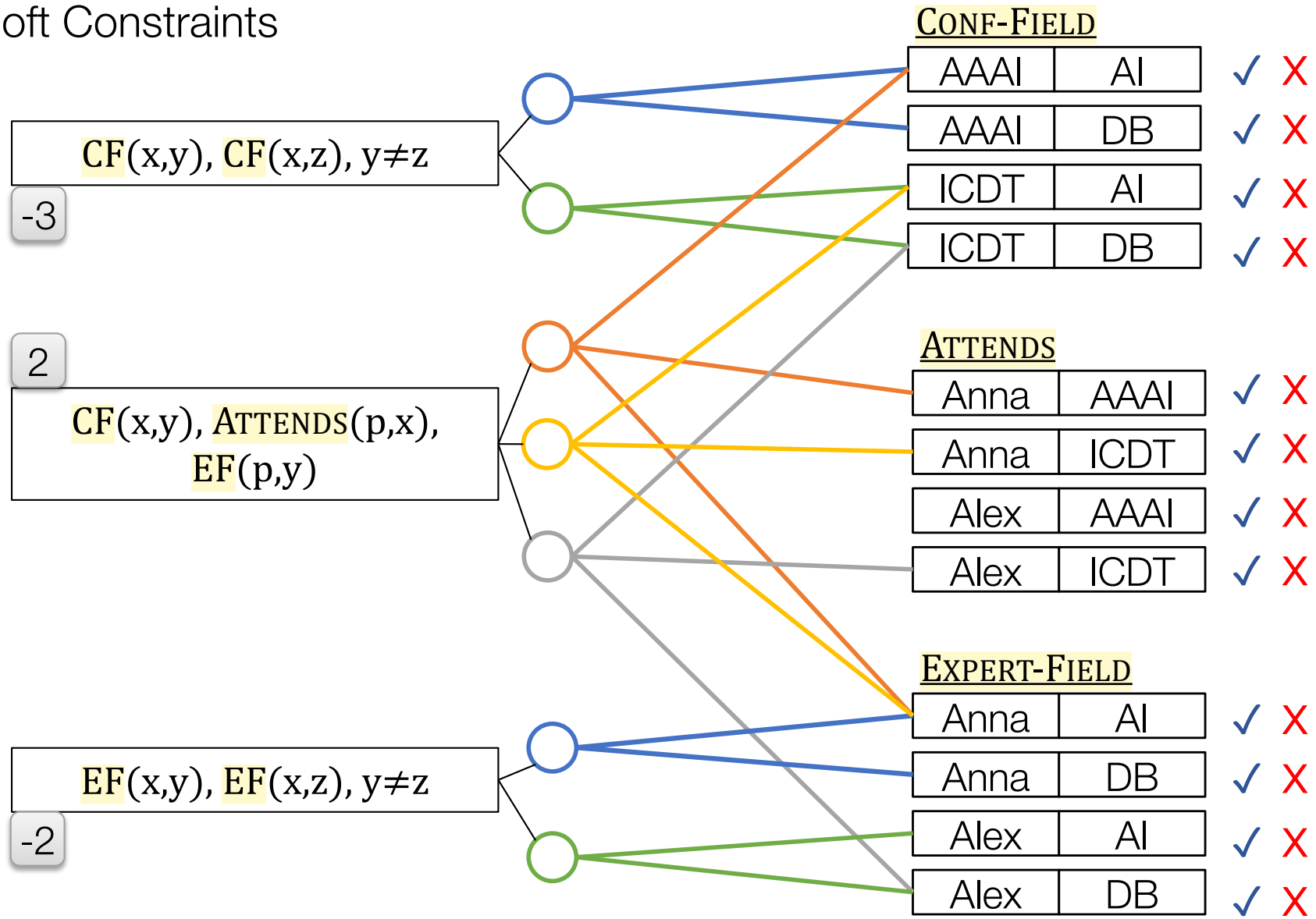
- Probabilistic variants of logic programming
  - The “distributional semantics” of Prob-Prog
  - Survey [DeRaedt-Kimmig15]
- Facts / ground rules drawn randomly; program applied deterministically

  [0.6]S(a,c). [0.8]R(a,b) :- S(a,c) , T(b,c).

  - Prism [Sato95, Sato-Kameya97]
  - Probabilistic Datalog [Rölleke-Fuhr97, Fuhr00]
  - ProbLog [DeRaedt-Kimmig-Toivonen07]
  - Independent Choice Logic [Poole08]

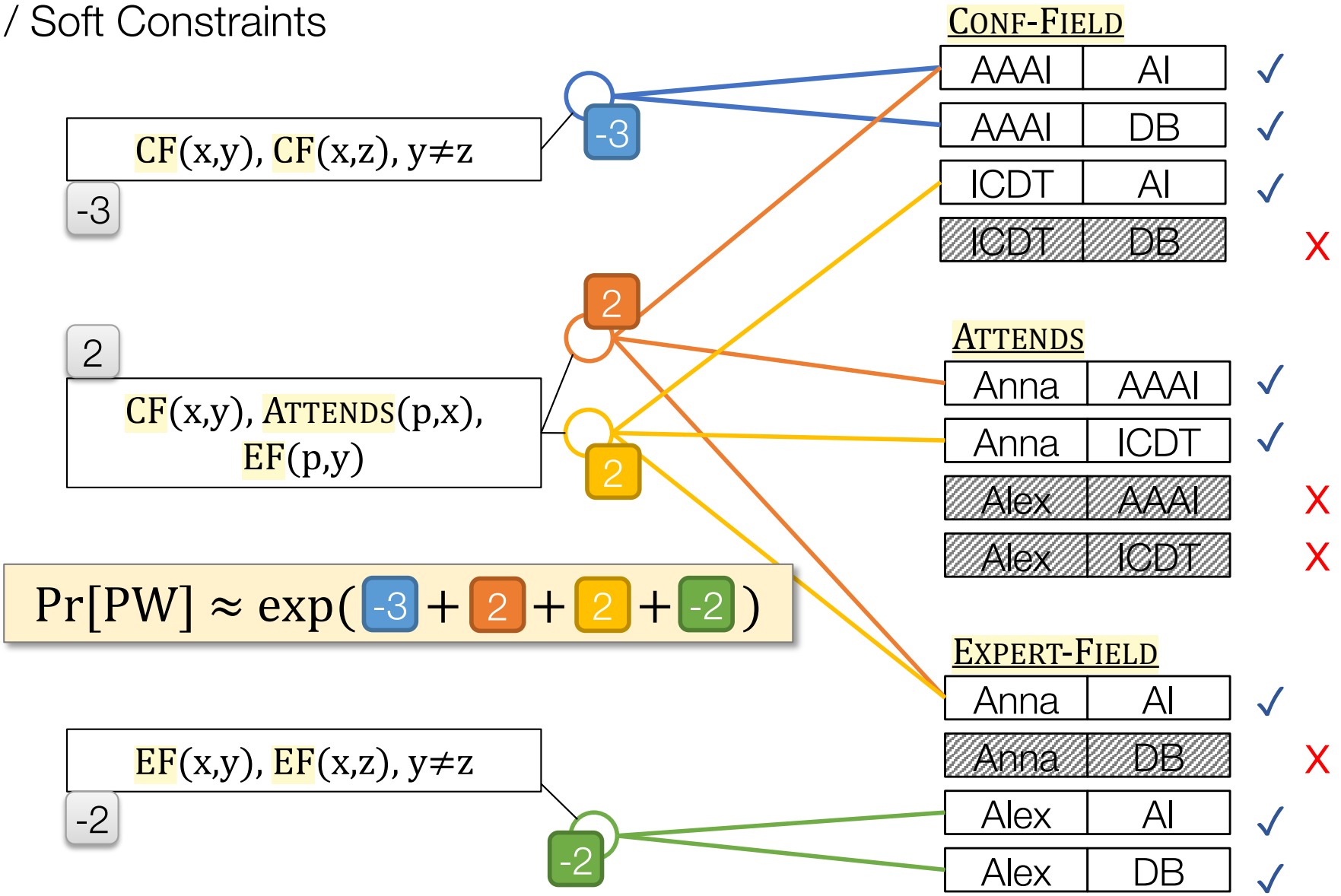
Tuple-Dependent Model

Markov Logic Networks [Richardson-Domingos06]
/ Soft Constraints



Tuple-Dependent Model

Markov Logic Networks [Richardson-Domingos06]
/ Soft Constraints



MLN vs TID

Correlations via few specific patterns simulated via expressive queries over independent tuples!

MLN more powerful than TID; complex correlations, not independence! **Wrong**

[Van den Broeck+11] [Jha-Suciu12] Inference over MLN can be translated into query evaluation over TID:

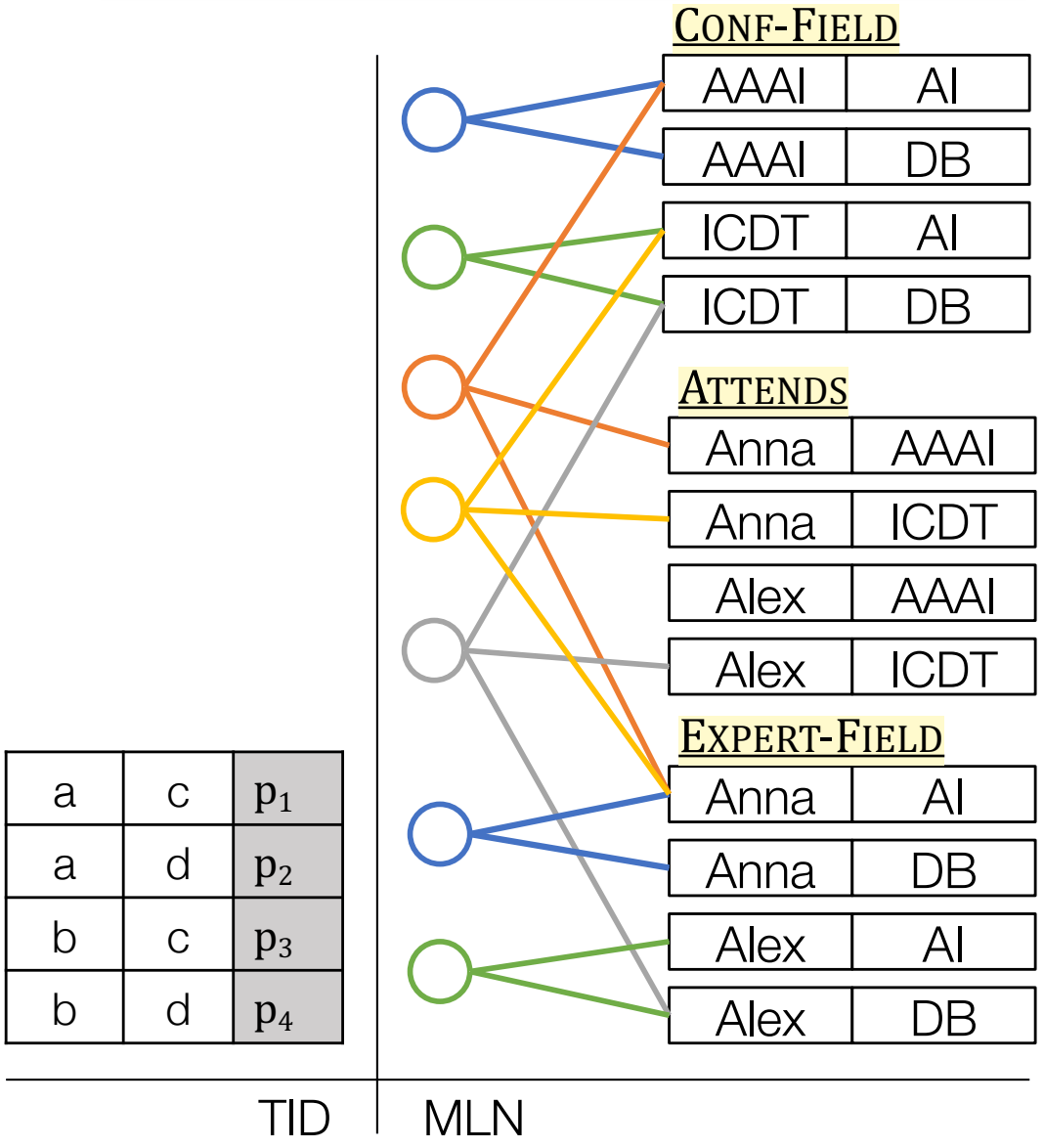
$$\Pr_{MLN}(Q) = \Pr_{TID}(Q,\Delta) \,/\, \Pr_{TID}(\Delta)$$

Queries (DB-ind.)

Caveats:

(a) Expressive queries (∀∃…)

(b) Require exact probs / multiplicative approx



The Database Complexity Perspective

- DB community: investigation through the lenses of **data complexity**
  - Separating query from data for complexity – every query is a distinct computational problem

- [Grädel-Gurevich-Hirsch98]: for the following Boolean query, marginal inference is #P-hard:

$$Q() \text{ :- } R(x), S(x,y), T(y,z), R(z)$$

Shorthand for $\exists x,y,z\ [R(x) \wedge S(x,y) \wedge T(y,z) \wedge R(z)]$

Overcoming the Hardness

- Statistical estimators
  - Monte Carlo (sample & average), Karp-Luby estimator, importance sampling [Grädel-Gurevich-Hirsch98] [Ré-Dalvi-Suciu07] [Gribkoff-Suciu16]
- Compilation into Boolean formulas and circuits [Olteanu-Huang08] [Jha-Suciu11]
- Syntactic relaxations of probability formulas
  - *Dissociation* – replace recurring variables w/ fresh ones [GatterbauerSuciu17] [VanDenHeuvel-Ivanov-Gatterbauer-Geerts-Theobald19]
  - *Read-once* approximations [Fink-Huang-Olteanu13]

Fundamental Dichotomy Theorem

THEOREM [Dalvi-Suciu04]

Let Q be a CQ without self-joins.

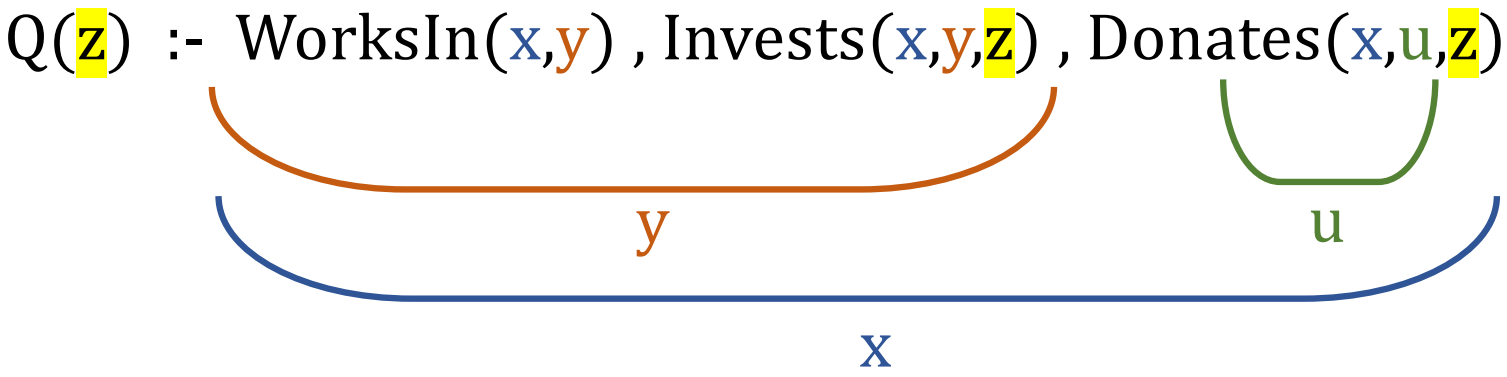
- If Q is **hierarchical**, then the probability of an answer can be computed in polynomial time.
- Otherwise, #P-complete.

Generalization to UCQs with self-joins [Dalvi-Suciu12]

Hierarchical Queries

A CQ (Conjunctive Query) is **hierarchical** if for every pair x and y of existential variables:

- $\mathbf{Atoms}(x) \subseteq \mathbf{Atoms}(y)$
- $\mathbf{Atoms}(y) \subseteq \mathbf{Atoms}(y)$
- $\mathbf{Atoms}(y)$ and $\mathbf{Atoms}(y)$ are disjoint



Quite remarkably, it turned out to characterize a plethora of fundamental DB properties

Characterizations of Hierarchical CQs

- There is a **safe plan** via the natural probability-aware version of Relational Algebra [Dalvi-Suciu04]
- The lineage of every answer is **read-once** [Olteanu-Huang08]
  - i.e. ≡ proposition where each var appears once
  - Hence, efficient inference [Gurvich77]
- *"How many DB subsets satisfy the query?"* can be computed in polynomial time [Amarilli-K20]
- The answer can be **updated instantly** following a DB update [Berkholz-Keppeler-Schweikardt17]
  - For non-Boolean CQs, adjustment needed
- Tractability of responsibility ⟶

Tuple's Responsibility to a Query

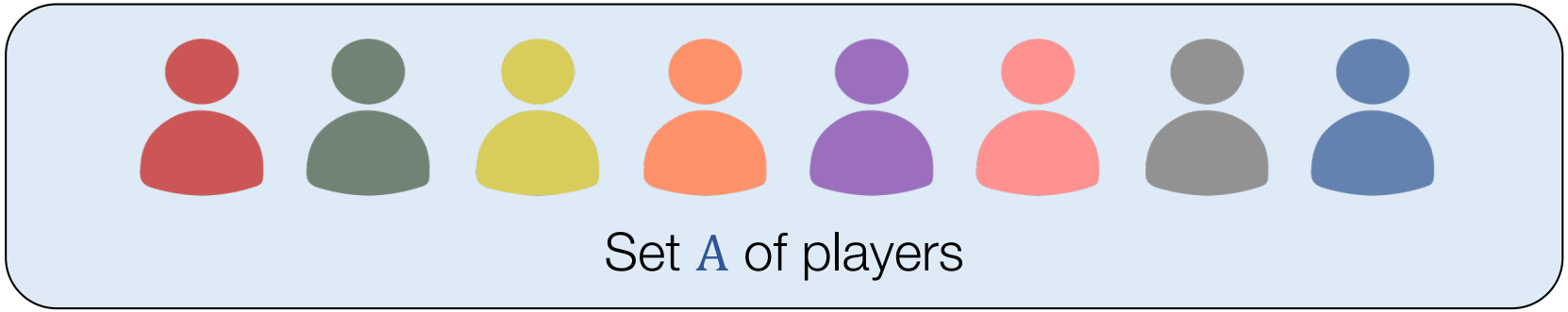
- *Which DB tuples explain a query answer?* Quantify each tuple's responsibility
- Various proposals
  - Counterfactual analysis [Meliou+10] [Freire+15]
    - Based on causality analysis: "*… minimal number of changes […] to obtain a contingency where B counterfactually depends on A*" [Chockler-Halpern04] ; How to generalize to aggregate queries?
  - Causal effect [Salimi-Bertossi-Suciu-VanDenBroeck16]
    - $\mathbb{E}[Q|\,\text{tuple}] - \mathbb{E}[Q|\,\neg\text{tuple}]$ when the DB is considered a TID
    - Similar to earlier ideas [Kanagal-Li-Deshpande11]
    - Called "Banzhaf Power Index" in cooperative game theory
- Problem long studied in coop. game theory: *how to distribute wealth among the team players?*
  - Most notably, the **Shapley value**

The Shapley Value

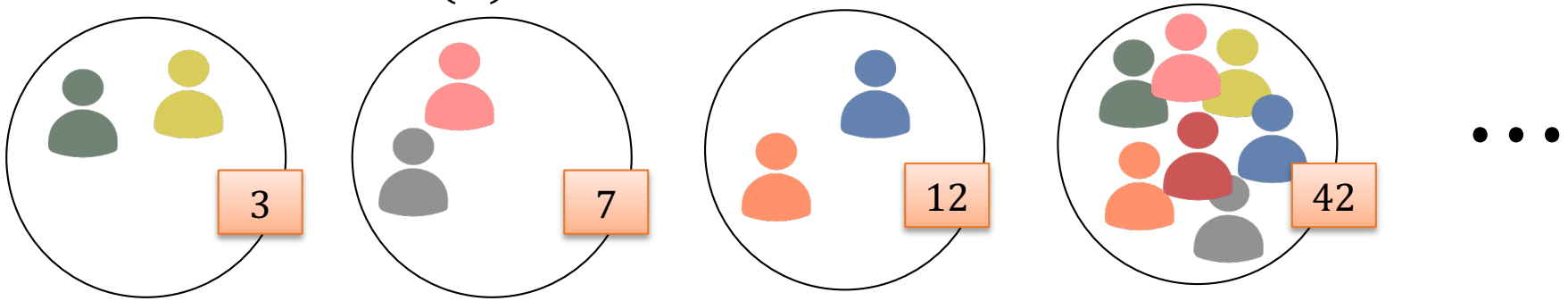
- A widely known profit-sharing formula in cooperative game theory by Shapley [1953]
- Theoretical justification: **unique modulo rationality desiderata**
- Applied in various areas:
  - Pollution responsibility in environmental management
  - Influence measurement in social network analysis
  - Identifying candidate autism genes
  - Bargaining foundations in economics
  - Takeover corporate rights in law
  - *Local explanations* in machine learning

[L.S. Shapley: Stochastic Games, 1953]

Shapley Definition



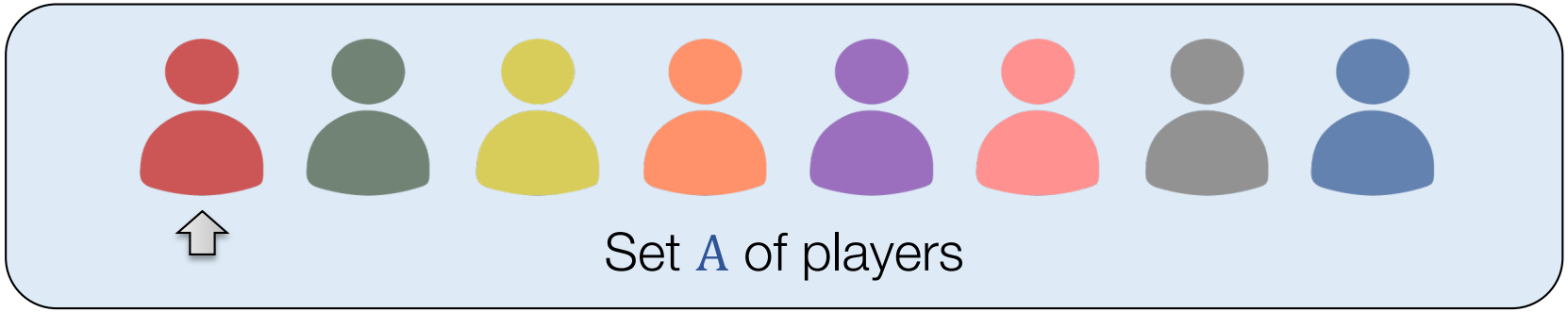
Wealth function $v: \mathcal{P}(A) \longrightarrow \mathbb{R}$

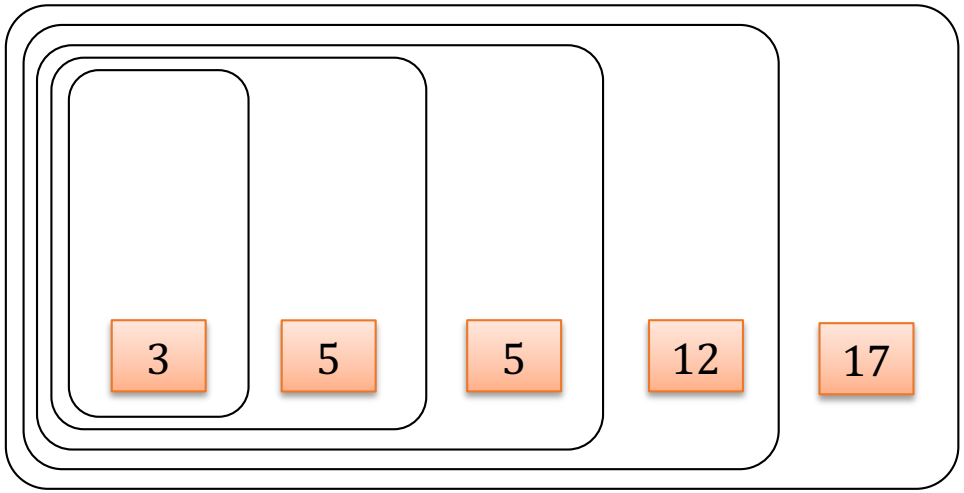


How to share the wealth among the players?

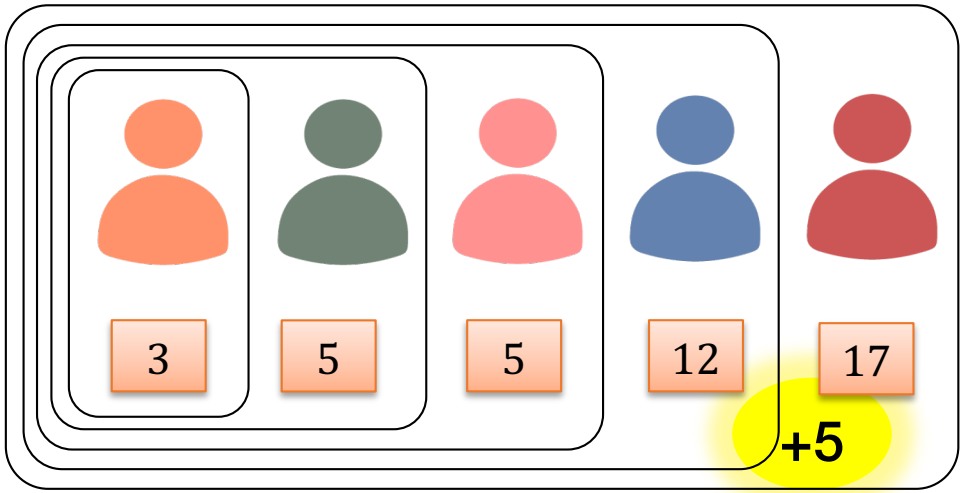
$$\text{Shapley}(A, v, a) = \sum_{B \subseteq A \setminus \{a\}} \frac{|B|!\,(|A| - |B| - 1)!}{|A|!} \big(v(B \cup \{a\}) - v(B)\big)$$

Shapley Explained



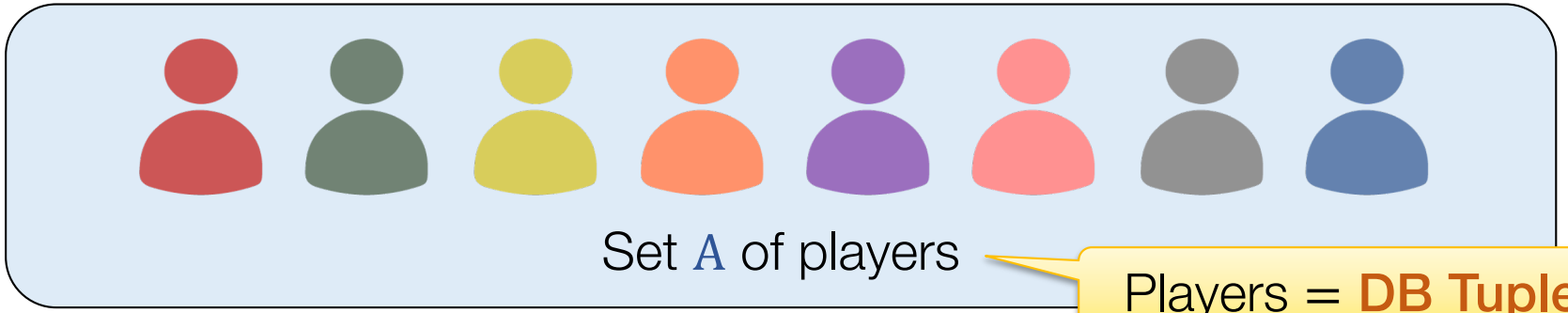


Shapley Explained

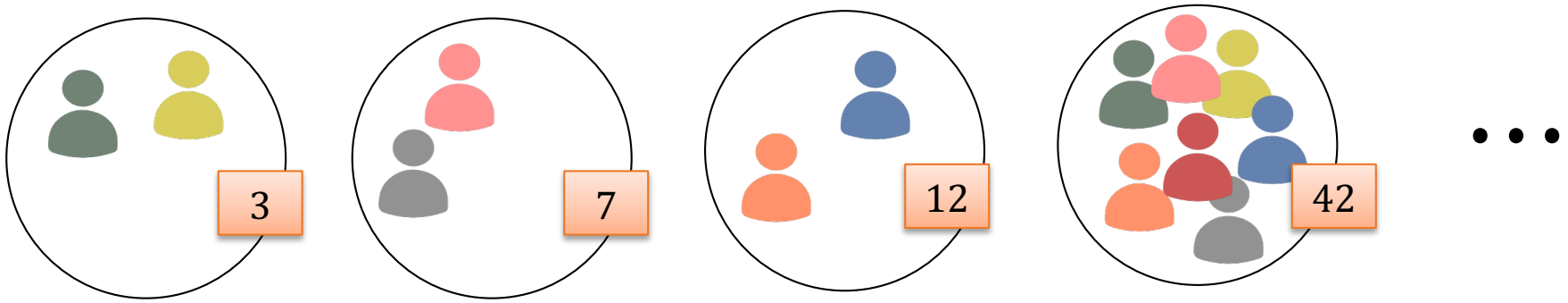


Shapley value: expected delta due to the addition in a *random permutation*

Shapley Definition for Database Queries



Wealth function $v: \mathcal{P}(A) \longrightarrow \mathbb{R}$ $v(B) = \text{Query}(B)$



$$\text{Shapley}(A, v, a) = \sum_{B \subseteq A \setminus \{a\}} \frac{|B|!\,(|A| - |B| - 1)!}{|A|!} \left(v(B \cup \{a\}) - v(B)\right)$$

[Livshits-Bertossi-K-Sebag ICDT2020] [Reshef-K-Livshits PODS2020]

Shapley Value for Database Queries

- The Shapley value is applicable to every function that maps DBs to numbers
  - Boolean, non-monotonic, aggregate queries, parameter learning in ML, inference in ML, …
- *What is the tuple's Shapley value for a CQ?*
  - [Livshits-Bertossi-K-Sebag ICDT2020]
  - [Reshef-K-Livshits PODS2020]

THEOREM [Livshits-Bertossi-K-Sebag2020]

For a CQ w/o self-joins, these are equivalent:

- The probability of an answer can be computed in polynomial time over **TID**s (i.e., Q is **hierarchical***).
- The **Shapley value** of a tuple can be computed in polynomial time.

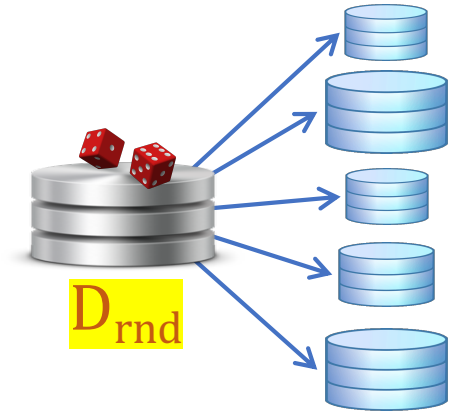
Hence, Shapley value gives another characterization of the hierarchical CQs!

More details in ICDT Session 3 (tomorrow)

“Probabilistic Databases” beyond Tuple Independence

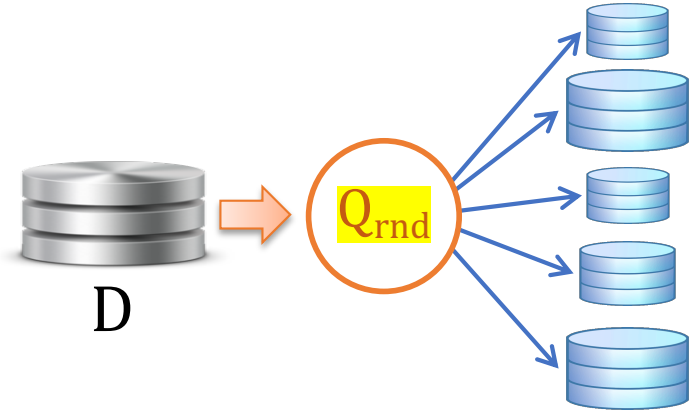
PDB types as different randomized interventions in query answering

Where is the Randomness?



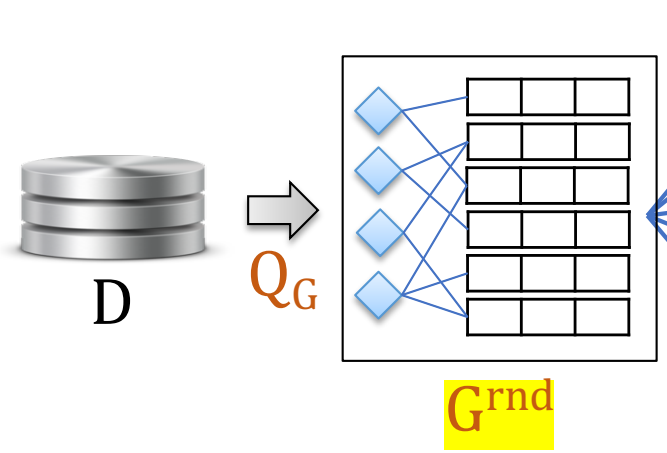
Pr(Q)

Source-Generative



Pr(Q)

Query-Generative



Pr(Q)

Target-Generative

Tuple-Independent DB

- Every tuple is annotated with a probability
- Independence:

$$\Pr[W] = \prod_{t \in W} p(t) \times \prod_{t \notin W} (1 - p(t))$$

person	city	state	p
Cullen	LA	CA	0.6
Cullen	Tampa	FL	0.4
Marion	LA	CA	1.0
Irene	NYC	NY	0.3
Irene	LA	FL	0.4

person	qualification	p
Cullen	9	0.3
Cullen	5	0.7
Marion	8	1.0
Irene	9	0.8

[deRougemont95] [Grädel-Gurevich-Hirsch98] [Dalvi-Suciu04]

Generalizations

person	city	state	
Cullen	LA	CA	0.6
Cullen	Tampa	FL	0.4
Marion	LA	CA	1.0
Irene	NYC	NY	0.3
Irene	LA	FL	0.4

Choose 1

Choose 0/1

Block-Independent DB (**BID**)
[Dalvi-Ré-Suciu11]

Random assignment

person	city	state	
Cullen	LA	CA	x=1
Cullen	Tampa	FL	x=2, y=1
Marion	LA	CA	
Irene	NYC	NY	y=1
Irene	LA	FL	y=3

U-relations
[Antova+07]

Condition over random assignment

person	city	state	
Cullen	LA	x	x=CA ∧ y≠CA
Cullen	Tampa	y	y=FL ∨ x≠y
Marion	LA	CA	
Irene	NYC	NY	y=1
Irene	LA	y	y=3

Probabilistic conditional (**pc-tables**)
[Green-Tannen06]

Preference Database

Candidates

cand	DoB	PoB
Bernie	1941	NY
Elizabeth	1949	OK
Joe	1942	PA

Voters

voter	DoB
Susan	1981
David	2000
James	1943

Pref

pole	voter	rank
p1	Susan	B < E < J
p1	David	E < J < B
p2	David	J < E < B

Q(v) :- Pref('p1',v,x,y),
Cnd(x, _,'NY'), Cnd(y, 'OK')

Q() :- Winner(v), Cnd(v,y,s), y<1945

- Querying election outcomes

Q(v) :- Winner(v)

- Winner determination
- By which *voting rule*?

Source-Generative Preference Database

Candidates

cand	DoB	PoB
Bernie	1941	NY
Elizabeth	1949	OK
Joe	1942	PA

Voters

voter	DoB
Susan	1981
David	2000
James	1943

Pref

pole	voter	rank
p1	Susan	Stat. Model
p1	David	Stat. Model
p2	David	Stat. Model

Q(v) :- **Pref**('p1',v,x,y),
Cnd(x, _,'NY'), Cnd(y, 'OK')

Q() :- **Winner**(v), Cnd(v,y,s), y<1945

- Querying election outcomes

Q(v) :- **Winner**(v)

- Winner determination
- By which *voting rule*?

Example: Mallows (1957)

Parameters: reference ranking ρ , dispersion ϕ

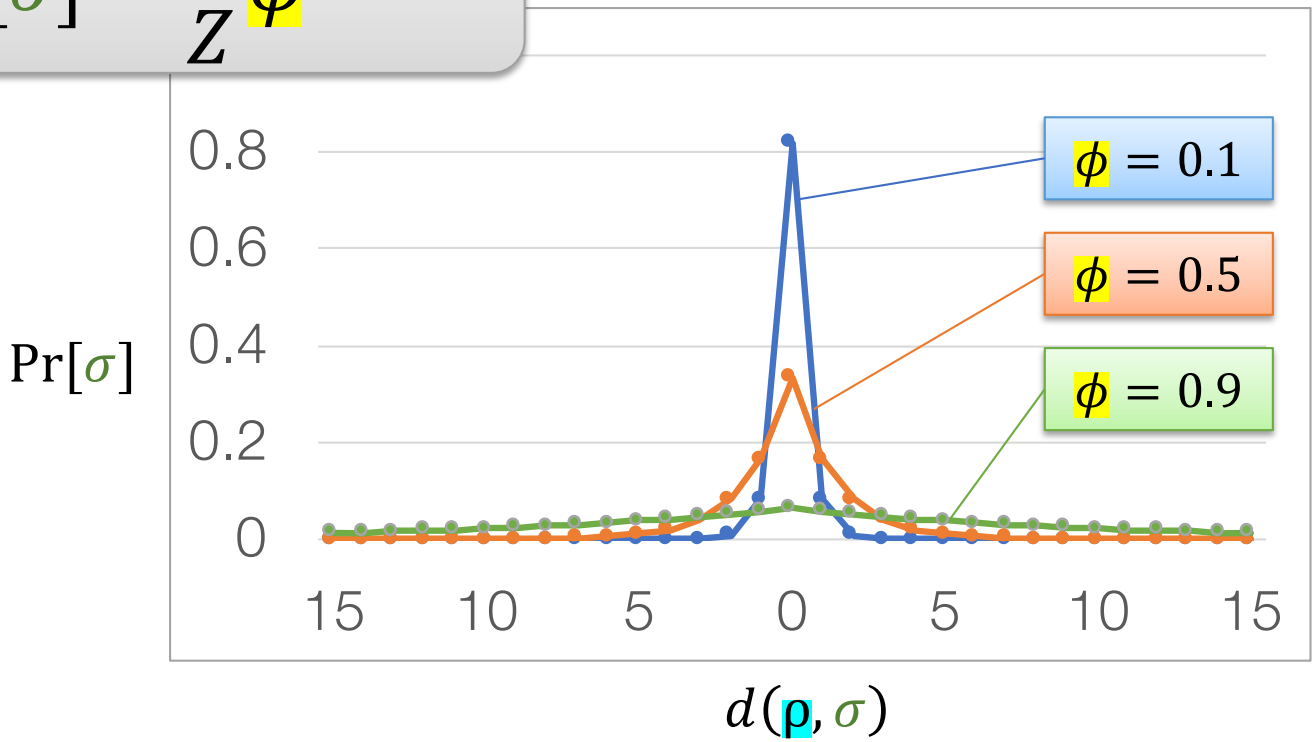
$$\rho = [a_1, a_2, \ldots, a_n] \qquad \phi \in (0,1]$$

$$\sigma = [a_{i_1}, a_{i_2}, \ldots, a_{i_n}]$$

$$\Pr[\sigma] = \frac{1}{Z}\phi^{d(\rho,\sigma)}$$

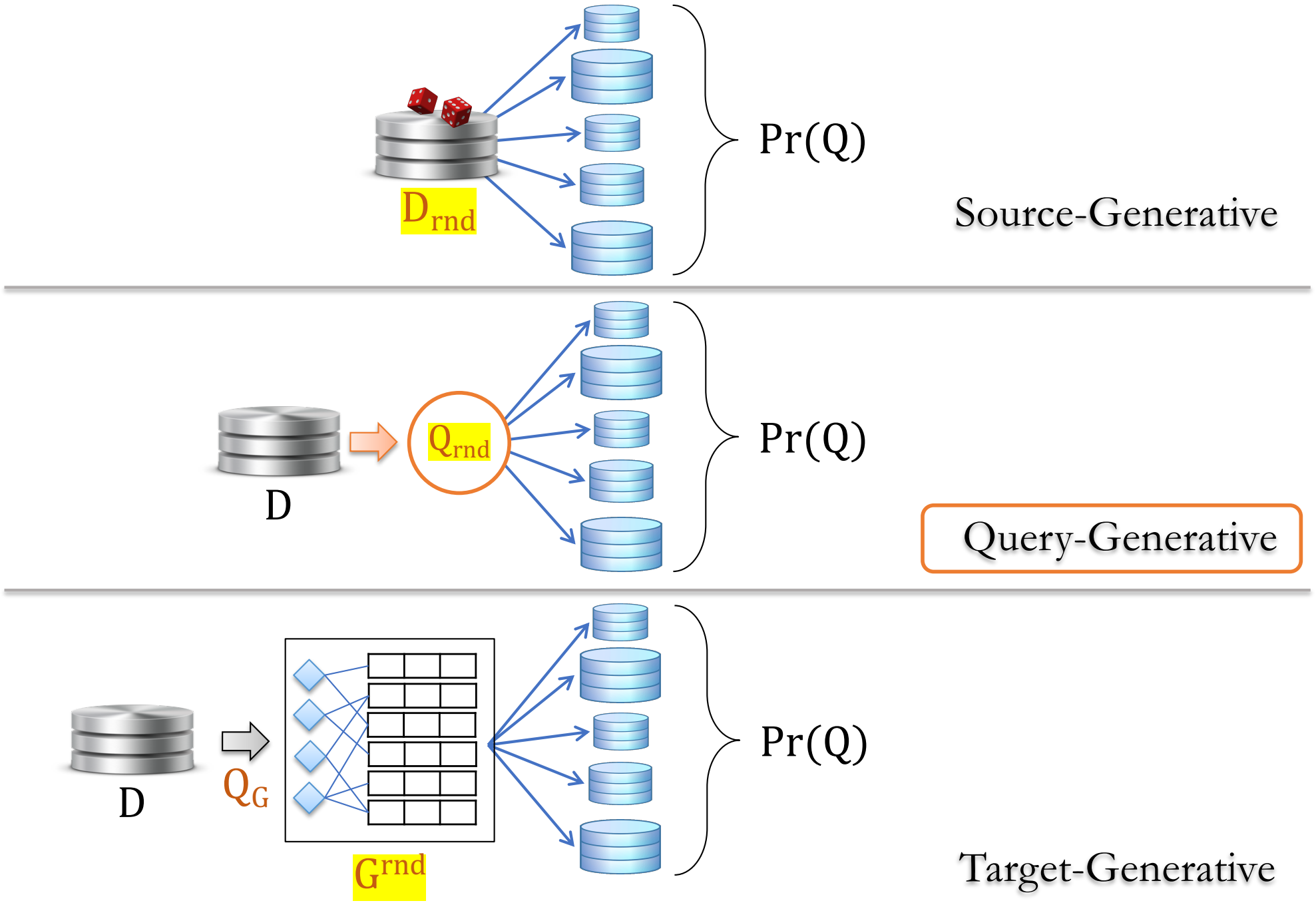
$$d(\rho, \sigma) = \sum_{j<k} \mathbf{1}[i_j > i_k]$$

(Kendal-tau distance)



Studied Problems

- Query evaluation on prob. preference DBs
  - [Ping-K-Stoyanovich VLDB20] [Kenig-Ilijasic-Ping-K-Stoyanovich AAAI18] [Cohen-Kenig-Ping-K-Stoyanovich SIGMOD18] [Kenig-K-Ping-Stoyanovich PODS17]
- Query evaluation on election outcomes over *partial preferences*
  - Only the extremes: possible / certain answers
  - "Computational Social Choice meets Databases"
  - [K-Kolaitis-Stoyanovich IJCAI18] [K-Phokion-Tibi PODS19]
- Winner determination over prob. preferences
  - [Kenig-K AAAI19]
- Open: Query evaluation on election outcomes over probabilistic preferences


D_{rnd}
Pr(Q)
Source-Generative
D
Q_{rnd}
Pr(Q)
Query-Generative
D
Q_G
G^{rnd}
Pr(Q)
Target-Generative

MCDB [Jampani+08]

SimSQL: extension with recursion for stochastic simulations [Cai+13]

name	gender	bp
Marion	F	71.1
Cullen	M	73.3
Irene	F	67.6

random

```
CREATE TABLE blood_pressure(name, gender, bp) AS

FOR EACH p IN patients

  WITH rbp AS Normal (
    SELECT b.mean, b.std
      FROM bp b
    WHERE b.gender = p.gender)

  SELECT p.name, p.gender, r.value
    FROM rbp r
```

rbp $\sim N(m, s)$

value
71.1

patients

name	gender
Marion	F
Cullen	M
Irene	F

bp

gender	mean	std
F	68.4	6.4
M	68.6	10.9

[Jampani-Xu-Wu-Perez-Jermaine-Haas08]

Other Formalisms

$\text{repair-key}_{a@p}(R)$

a	b	p
PAR	ROM	0.2
PAR	BER	0.8
LON	CPH	0.4
LON	PAR	0.6

SELECT A.a, C.c, conf() FROM
(**repair key** a in A **weight by** p) R1,
(**repair key** b in B **weight by** q) R2, …

MayBMS [Huang-Antova-Koch-Olteanu09]

a	b
PAR	ROM
LON	CPH

0.2×0.4

a	b
PAR	ROM
LON	PAR

0.2×0.6

a	b
PAR	BER
LON	CPH

0.8×0.4

a	b
PAR	BER
LON	PAR

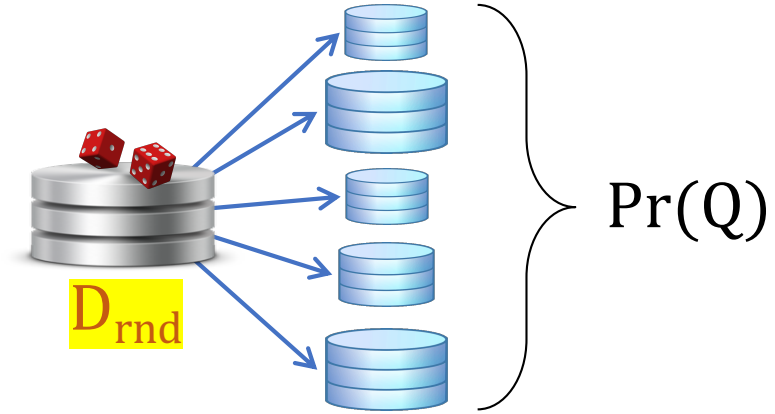
0.8×0.6

IsPromoted(pid,Flip[0.02]) ⟵ Products(pid,pname)

Sales(sid,pid,Poisson[r]) ⟵ Stores(sid,city), HistRate(city,pid,r), IsPromoted(pid,0)

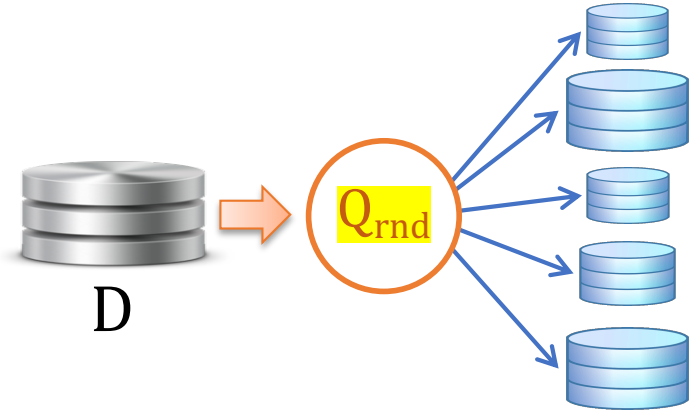
Sales(sid,pid,Poisson[q]) ⟵ Stores(sid,city), HistPrmRate(city,pid,q), IsPromoted(pid,1)

Probabilistic-Programming Datalog [Barany-tenCate-K-Olteanu-Vagena17]



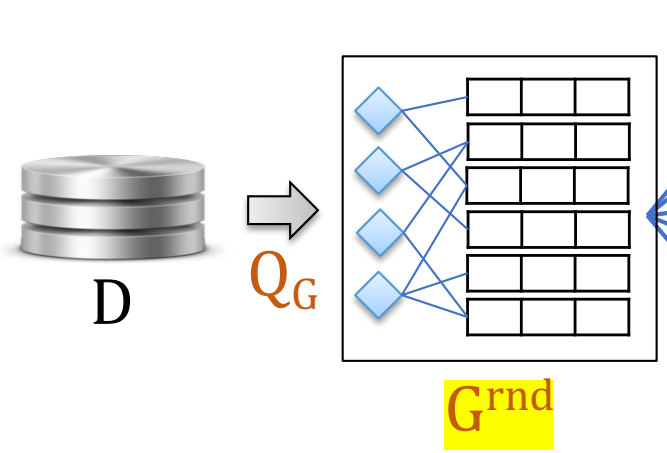
Pr(Q)

Source-Generative



Pr(Q)

Query-Generative



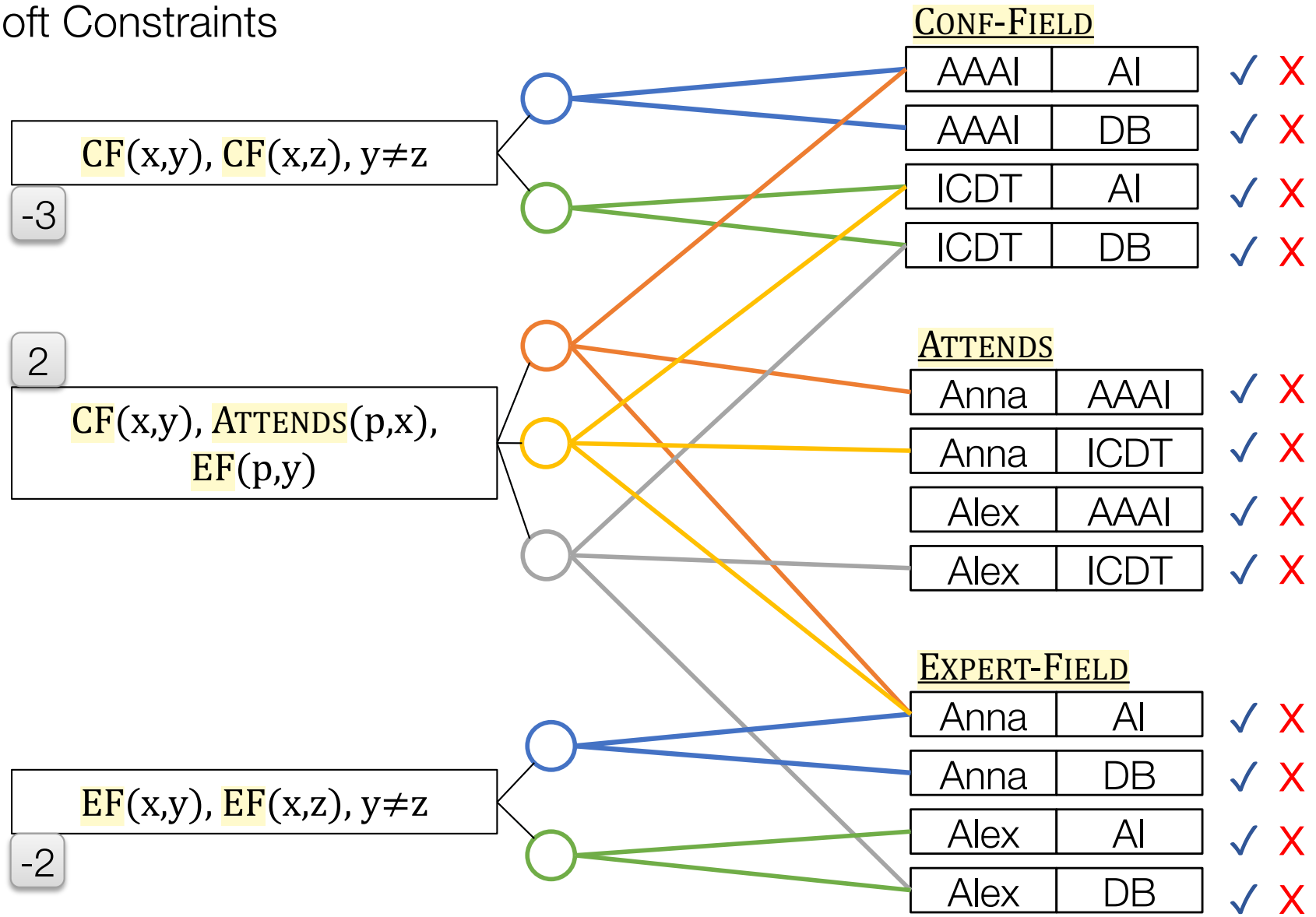
Pr(Q)

Target-Generative

MLN / Soft Constraints

Alchemy | DeepDive | Pr. DL+/- | HoloClean

Markov Logic Networks [Richardson-Domingos06] / Soft Constraints



The Perspective of Database Repairs

PDB modeling explains probabilistic approaches to DB repairing

Inconsistency and Repairs

- **Inconsistent database** violates constraints
  - Key/uniqueness constraints, functional dependencies, referential constraints, etc.
- **Repair**: a consistent variant via a *legitimate fix*
  - **Subset repairs**: set-max consistent subset
  - **Cardinality repairs**: cardinality-max consistent subset
  - Update repairs (value updates), symmetric-difference repairs (tuple insertion/deletion), …
  - [Arenas-Bertossi-Chomicki99]

Example: Subset/Cardinality Repairs

$$person \longrightarrow birthCity$$

$$birthCity \longrightarrow birthState$$

person	birthCity	birthState
Douglas	LA	CA
Douglas	Miami	FL
Tedrow	LA	CA
Tedrow	LA	NYC
Jones	LA	CA

person	birthCity	birthState
~~Douglas~~	~~LA~~	~~CA~~
Douglas	Miami	FL
~~Tedrow~~	~~LA~~	~~CA~~
Tedrow	LA	NYC
~~Jones~~	~~LA~~	~~CA~~

Subset repair

person	birthCity	birthState
~~Douglas~~	~~LA~~	~~CA~~
Douglas	Miami	FL
Tedrow	LA	CA
~~Tedrow~~	~~LA~~	~~NYC~~
Jones	LA	CA

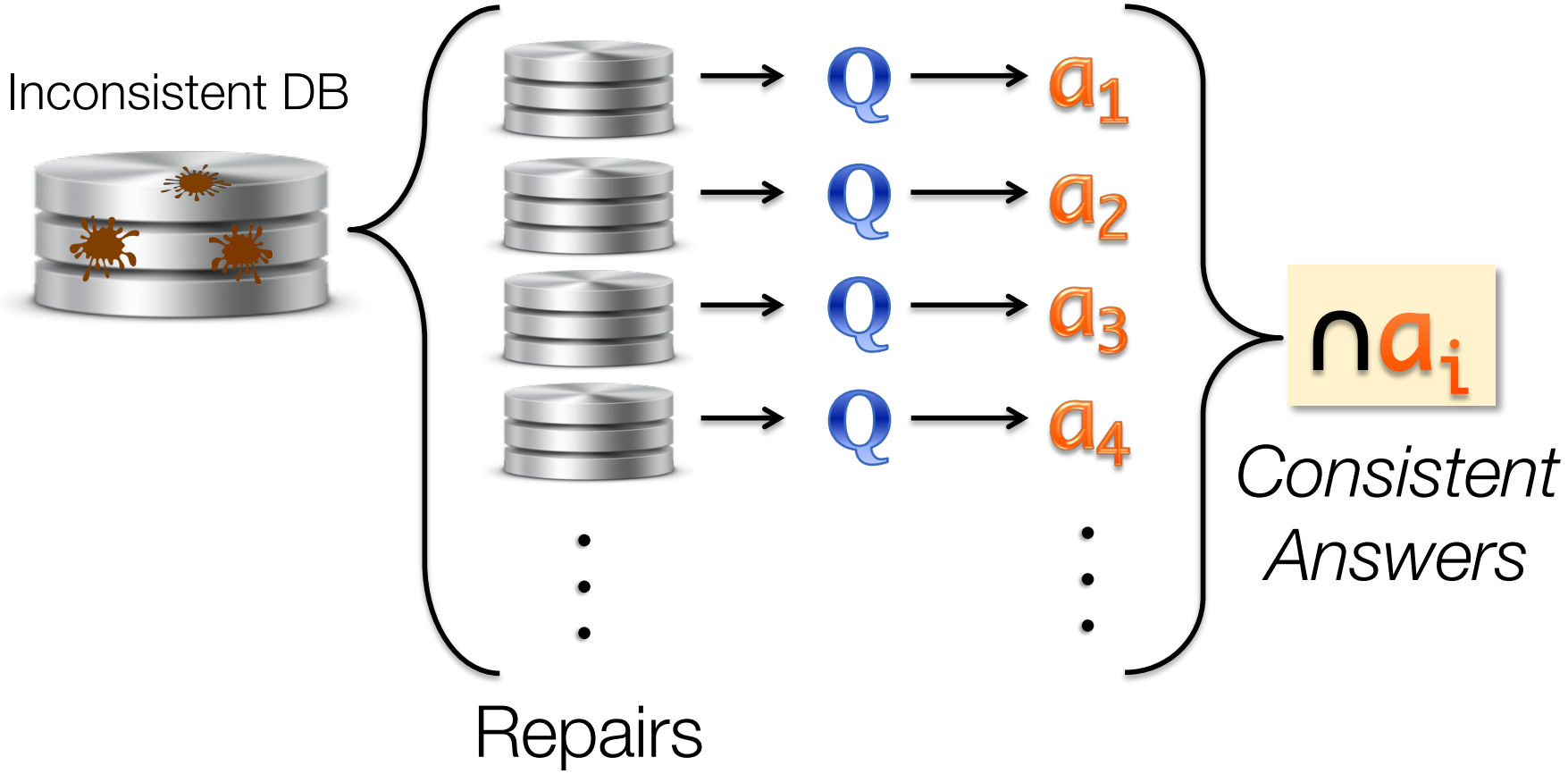
Cardinality (& subset) repair

Consistent Answers

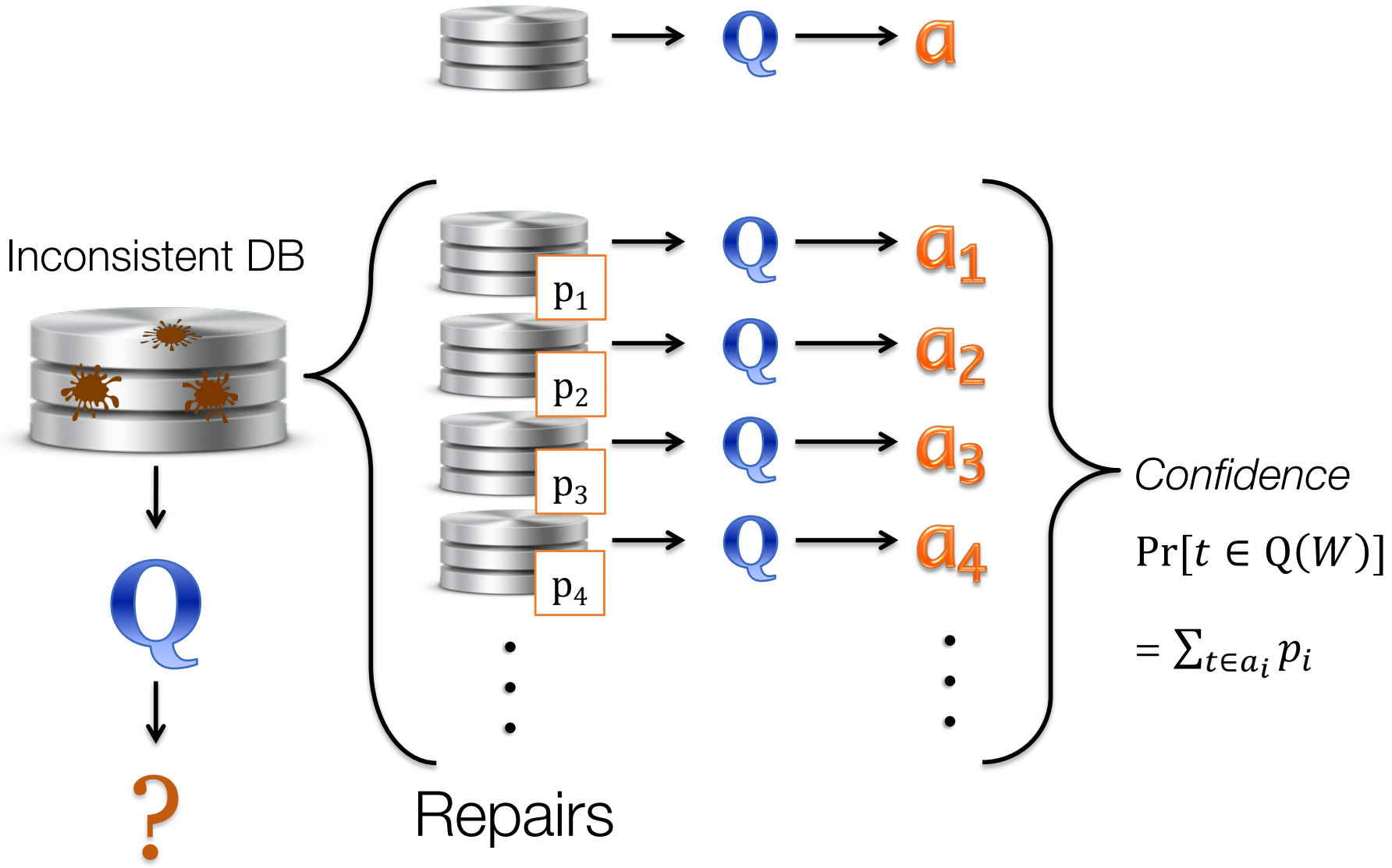
[Arenas-Bertossi-Chomicki99]

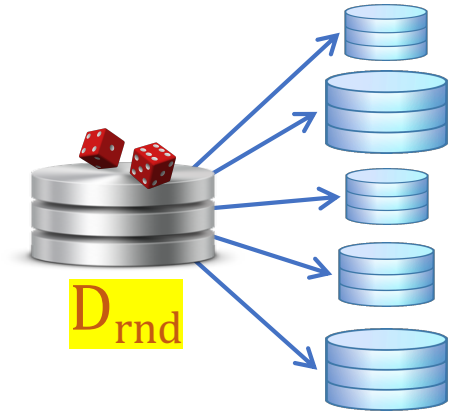
Difficulties:

- Hard to compute
- Overly restrictive
- No clear notion of approximation



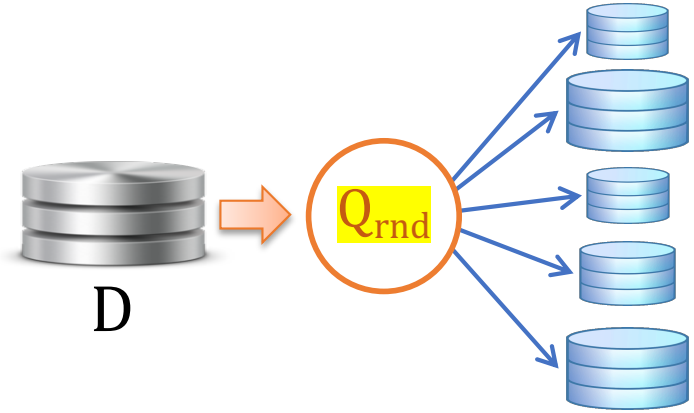
Probably Consistent Answers





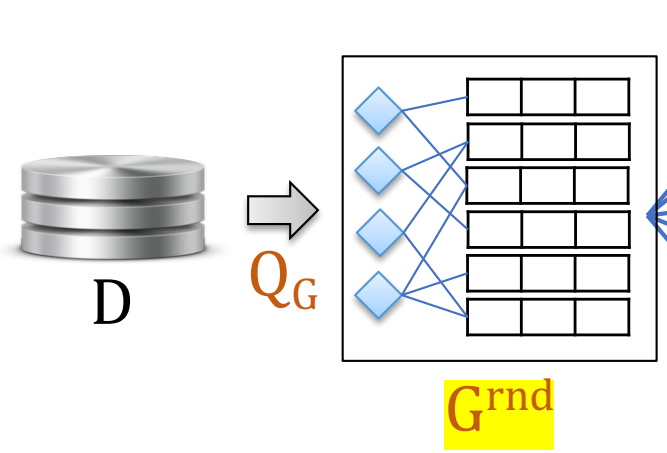
Pr(Q)

Source-Generative



Pr(Q)

Query-Generative



Pr(Q)

Target-Generative

Probabilistic Duplicates [Andritsos-Fuxman-Miller06]

person ⟶ birthCity, birthState

indep.

	person	birthCity	birthState	p
disjoint	Cullen Douglas	LA	CA	0.6
disjoint	Cullen Douglas	Tampa	FL	0.4
disjoint	Marion Jones	LA	CA	1.0
disjoint	Irene Tedrow	NYC	NY	0.3
disjoint	Irene Tedrow	LA	FL	0.4
disjoint	Irene Tedrow	Hollywood	FL	0.2
disjoint	Irene Tedrow	Hollywood	CA	0.1

Later termed **Block-Independent Databases** (BID) [Dalvi-Ré-Suciu11]

The uniform case: repair counting

[Greco-Molinaro12] [Maslowski-Wijsen14]

Beyond Key Constraints?

$$\text{person} \longrightarrow \text{birthCity}$$
$$\text{birthCity} \longrightarrow \text{birthState}$$

person	birthCity	birthState
Cullen Douglas	LA	CA
Cullen Douglas	Tampa	FL
Marion Jones	LA	CA
Irene Tedrow	NYC	NY
Irene Tedrow	LA	FL
Irene Tedrow	Hollywood	FL
Irene Tedrow	Hollywood	CA

Constrained TID [Gribkoff-VanDenBroeck-Suciu14]

$$\text{person} \longrightarrow \text{birthCity}$$
$$\text{birthCity} \longrightarrow \text{birthState}$$

person	birthCity	birthState	p
Cullen Douglas	LA	CA	0.6
Cullen Douglas	Tampa	FL	0.7
Marion Jones	LA	CA	0.9
Irene Tedrow	NYC	NY	0.6
Irene Tedrow	LA	FL	0.9
Irene Tedrow	Hollywood	FL	0.5
Irene Tedrow	Hollywood	CA	0.8

$$\mathrm{p}(W) = \Pr(W \mid C)$$

Basic question: most probable W (MPD)

MPD

person ⟶ birthCity
birthCity ⟶ birthState

factor	person	birthCity	birthState	*p*
0.6	Cullen Douglas	LA	CA	0.6
1-0.7	Cullen Douglas	Tampa	FL	0.7
1-0.9	Marion Jones	LA	CA	0.9
1-0.6	Irene Tedrow	NYC	NY	0.6
1-0.9	Irene Tedrow	LA	FL	0.9
1-0.5	Irene Tedrow	Hollywood	FL	0.5
0.8	Irene Tedrow	Hollywood	CA	0.8

$$\max_{\text{consistent } J}\left(\prod_{t\in J} p(t) \times \prod_{t\notin J}(1-p(t))\right)$$

MPD

person ⟶ birthCity
birthCity ⟶ birthState

factor	person	birthCity	birthState	*p*
1-0.6	~~Cullen Douglas~~	~~LA~~	~~CA~~	0.6
0.7	Cullen Douglas	Tampa	FL	0.7
0.9	Marion Jones	LA	CA	0.9
1-0.6	~~Irene Tedrow~~	~~NYC~~	~~NY~~	0.6
1-0.9	~~Irene Tedrow~~	~~LA~~	~~FL~~	0.9
1-0.5	~~Irene Tedrow~~	~~Hollywood~~	~~FL~~	0.5
0.8	Irene Tedrow	Hollywood	CA	0.8

Can compute efficiently?

$$\max_J \left(\prod_{t \in J} p(t) \times \prod_{t \notin J} (1 - p(t)) \right)$$

MPD is tightly close to a classic problem:

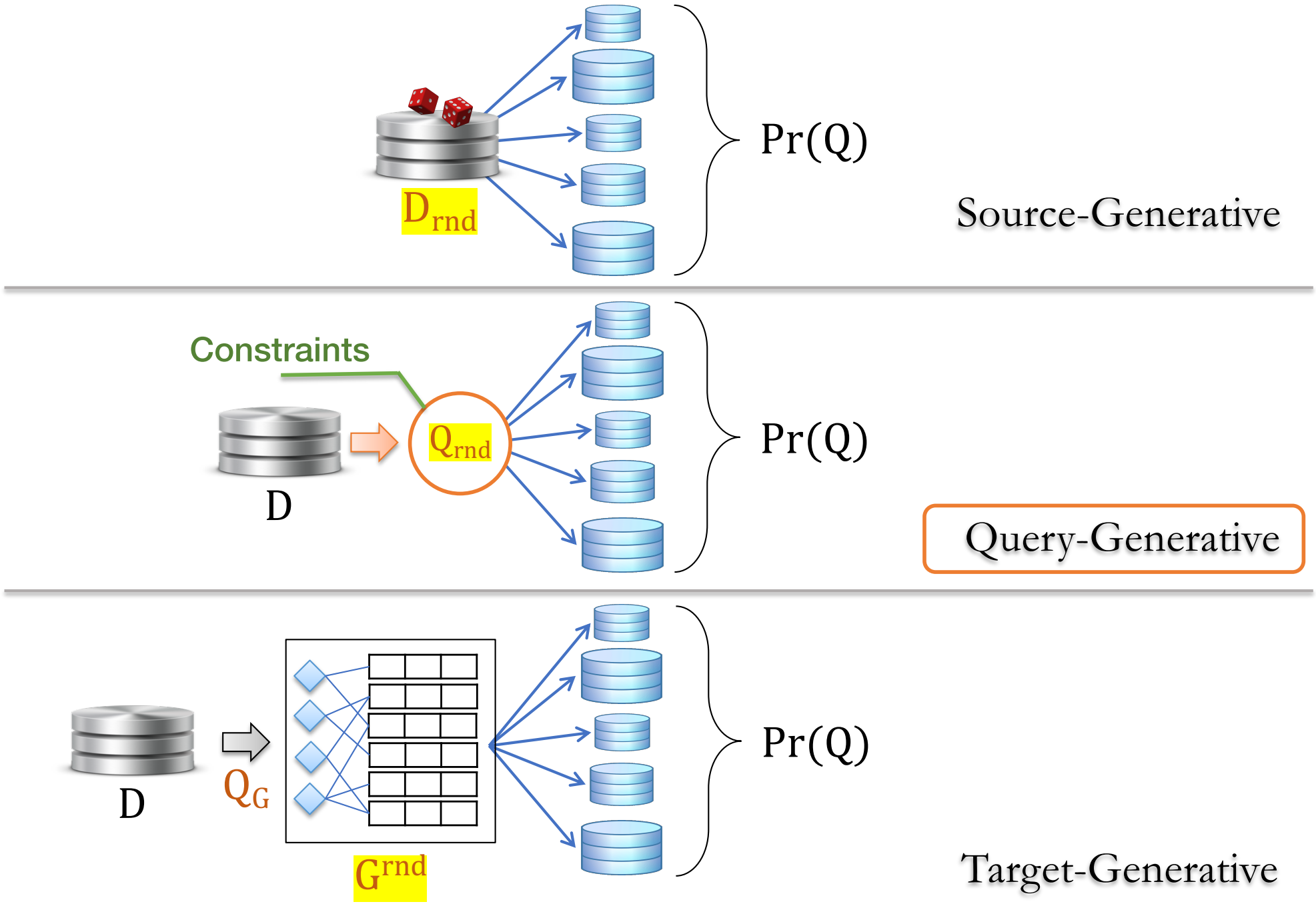
Compute a **cardinality repair**

THEOREM [Livshits-K-Roy2018]

Fix any set of FDs:

1. If a **cardinality repair** can be found in polynomial time, then so can an **MPD**.
2. Conversely, if **cardinality repair** is hard, then **MPD** *(or any approximation)* is hard.

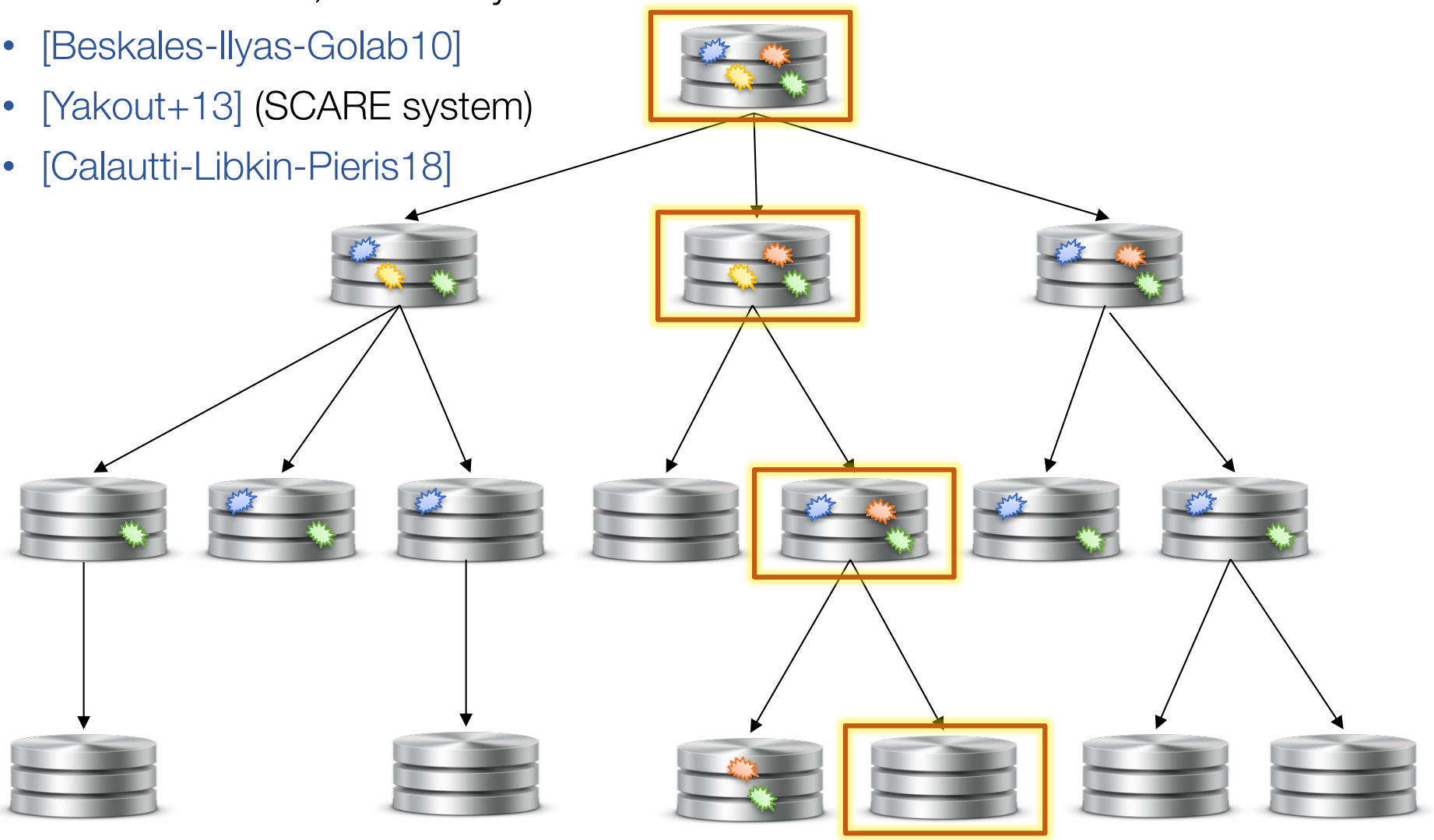
Moreover, we can test which of the two situations hold.


D_{rnd}
Pr(Q)
Source-Generative
Constraints
D
Q_{rnd}
Pr(Q)
Query-Generative
D
Q_G
G^{rnd}
Pr(Q)
Target-Generative

Repairing as Markov Chain

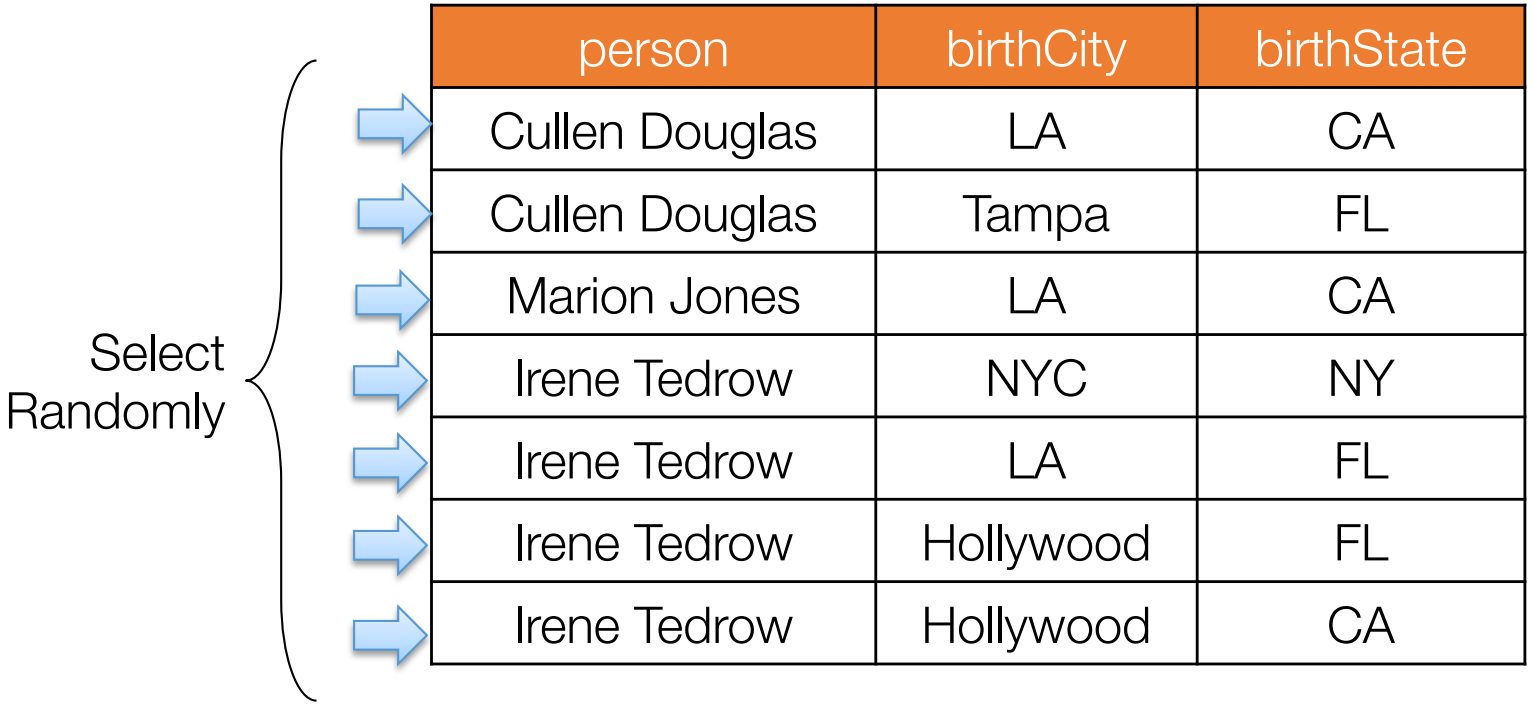
Idea: Iteratively select violations and fix, randomly

- [Beskales-Ilyas-Golab10]
- [Yakout+13] (SCARE system)
- [Calautti-Libkin-Pieris18]



Operational CQA [Calautti-Libkin-Pieris18]

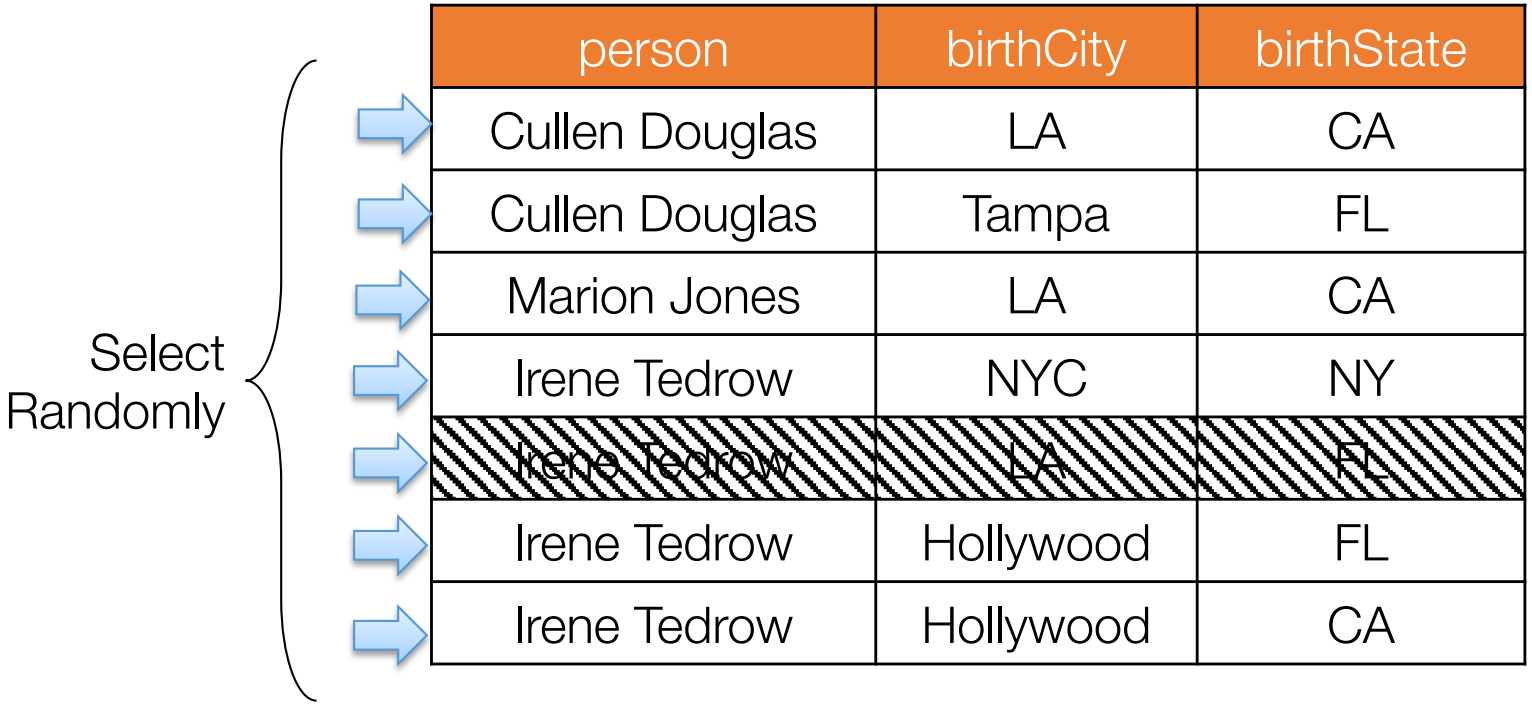
$$\text{person} \longrightarrow \text{birthCity}$$
$$\text{birthCity} \longrightarrow \text{birthState}$$



person	birthCity	birthState
Cullen Douglas	LA	CA
Cullen Douglas	Tampa	FL
Marion Jones	LA	CA
Irene Tedrow	NYC	NY
Irene Tedrow	LA	FL
Irene Tedrow	Hollywood	FL
Irene Tedrow	Hollywood	CA

Operational CQA

$$\text{person} \longrightarrow \text{birthCity}$$
$$\text{birthCity} \longrightarrow \text{birthState}$$



person	birthCity	birthState
Cullen Douglas	LA	CA
Cullen Douglas	Tampa	FL
Marion Jones	LA	CA
Irene Tedrow	NYC	NY
Irene Tedrow	LA	FL
Irene Tedrow	Hollywood	FL
Irene Tedrow	Hollywood	CA

[Calautti-Libkin-Pieris18]

Operational CQA [Calautti-Libkin-Pieris18]

$$\text{person} \longrightarrow \text{birthCity}$$
$$\text{birthCity} \longrightarrow \text{birthState}$$

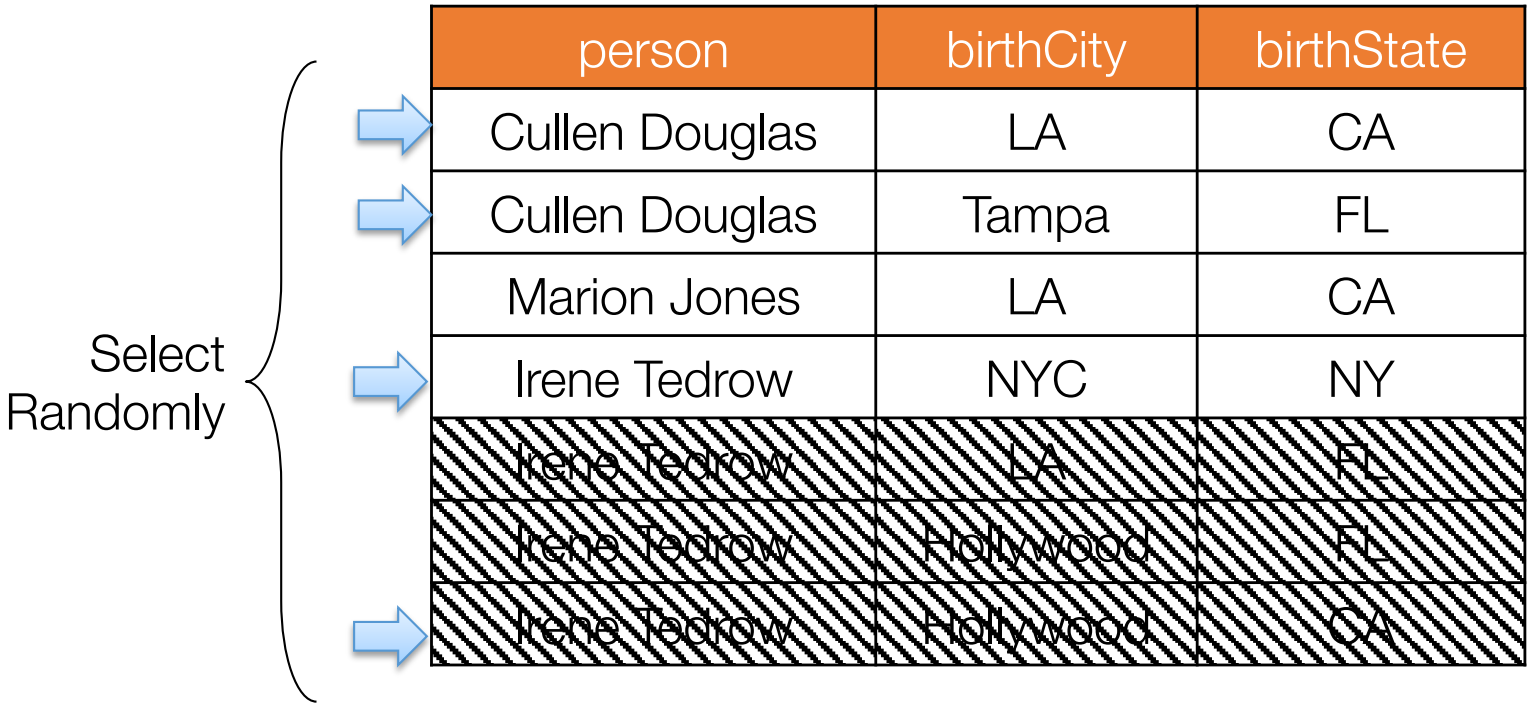
Select Randomly

person	birthCity	birthState
Cullen Douglas	LA	CA
Cullen Douglas	Tampa	FL
Marion Jones	LA	CA
Irene Tedrow	NYC	NY
Irene Tedrow	LA	FL
Irene Tedrow	Hollywood	FL
Irene Tedrow	Hollywood	CA

[Calautti-Libkin-Pieris18]

Operational CQA [Calautti-Libkin-Pieris18]

$$\text{person} \longrightarrow \text{birthCity}$$
$$\text{birthCity} \longrightarrow \text{birthState}$$



Operational CQA [Calautti-Libkin-Pieris18]

Generalizes to:

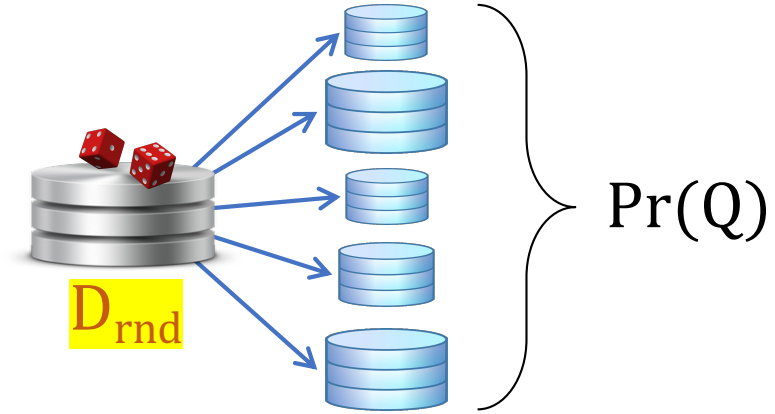
- Arbitrary constraints
- Tuple deletion/addition
- Tuple weights, etc.

$$\text{person} \longrightarrow \text{birthCity}$$
$$\text{birthCity} \longrightarrow \text{birthState}$$

Select Randomly

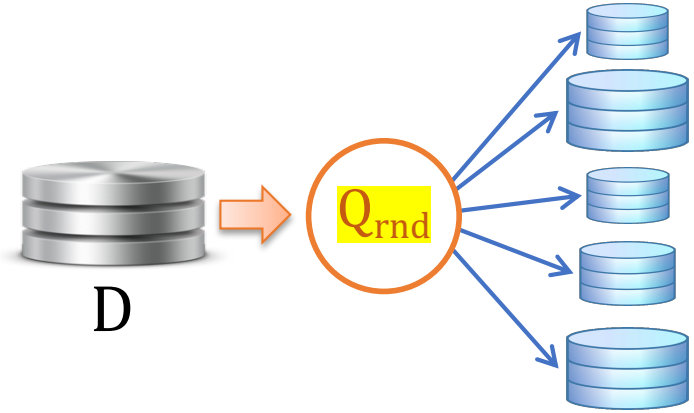
person	birthCity	birthState
Cullen Douglas	LA	CA
Cullen Douglas	Tampa	FL
Marion Jones	LA	CA
Irene Tedrow	NYC	NY
Irene Tedrow	LA	FL
Irene Tedrow	Hollywood	FL
Irene Tedrow	Hollywood	CA

[Calautti-Libkin-Pieris18]



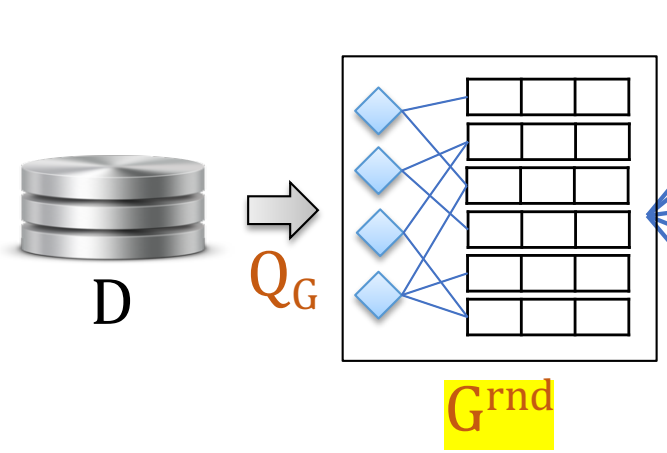
Pr(Q)

Source-Generative



Pr(Q)

Query-Generative



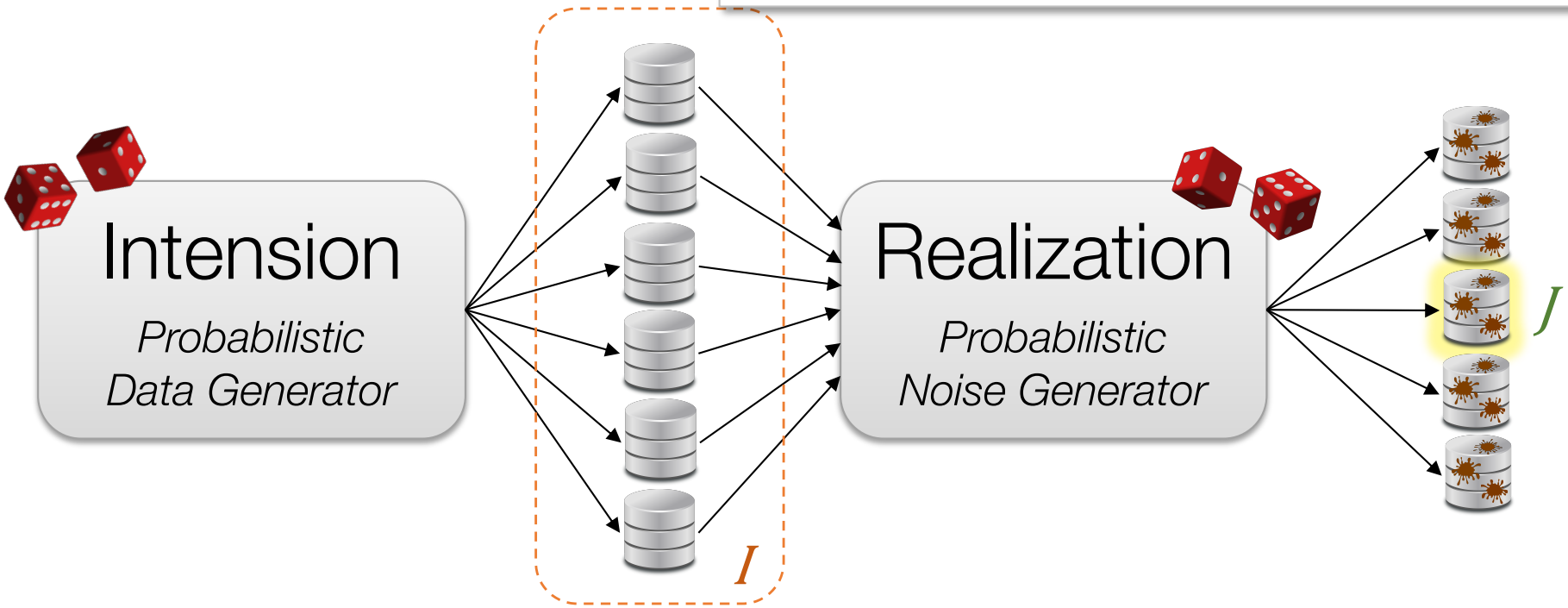
Pr(Q)

Target-Generative

PUD: Noisy Channel

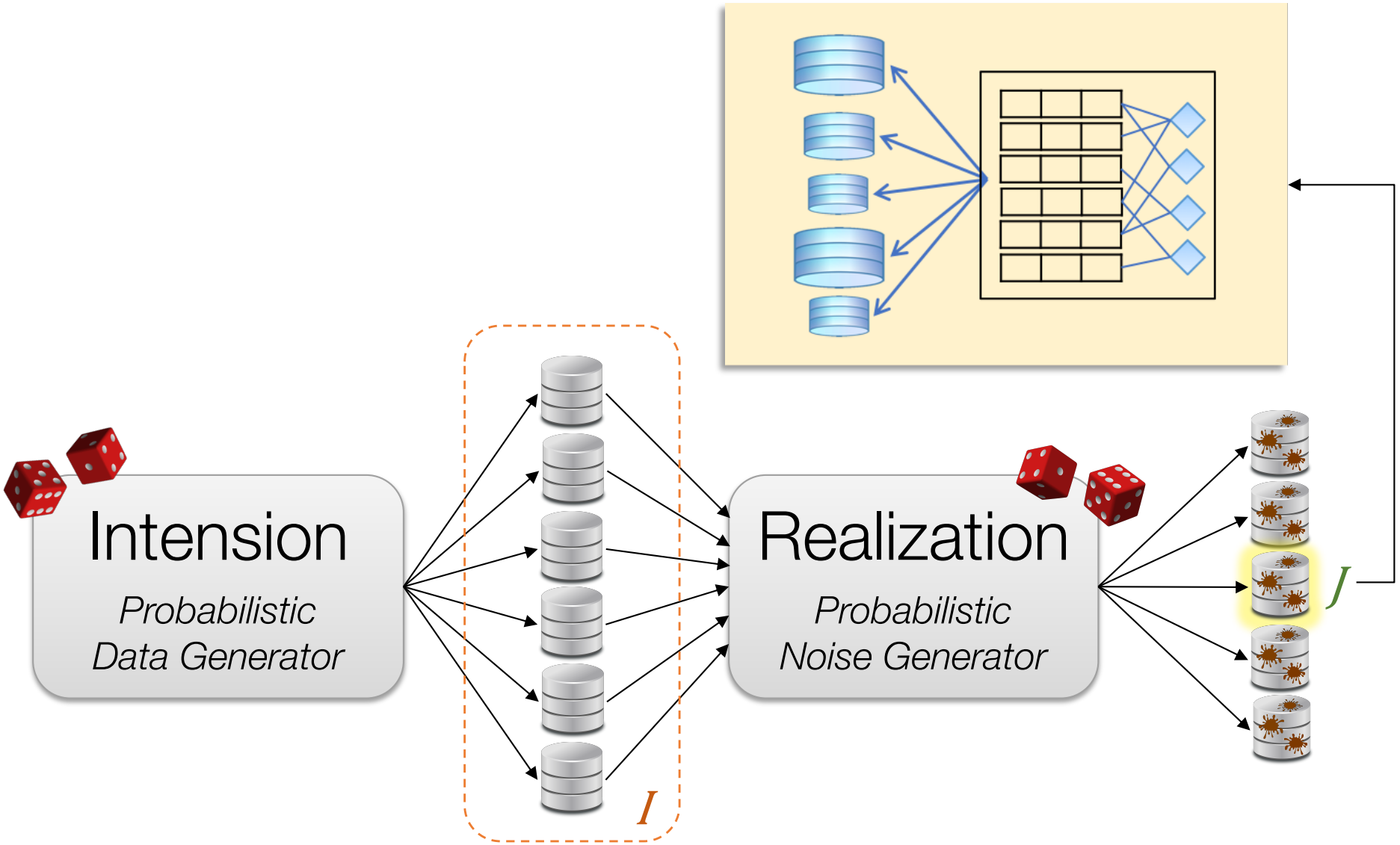
= Prob. Unclean Database

- HoloClean
  - [Rekatsinas-Chu-Ilyas-Ré17]
- HoloDetect
  - [Heidari-McGrath-Ilyas-Rekatsinas19]



[DeSa-Ilyas-K-Ré-Rekatsinas18]

Idea: Compile into One Big Factor Graph



[DeSa-Ilyas-K-Ré-Rekatsinas18]

Simple PUD Example

-50 $t_1.person = t_2.person$ & $t_1.birthCity \neq t_2.birthCity$

-5 $t_1.birthCity = t_2.birthCity$ & $t_1.birthCountry \neq t_2.birthCountry$

$$\text{Prob}(I) \sim \exp(\Sigma\ \text{penatlies}(I))$$

Intention

Probabilistic Data Generator

person	birthCity	birthCountry	
Douglas	LA	USA	-50
Douglas	Tampa	USA	-50
Khan	Ghajar	Lebanon	-5
Khan	Ghajar	Israel	-50
Khan	NYC	USA	-50

I

J

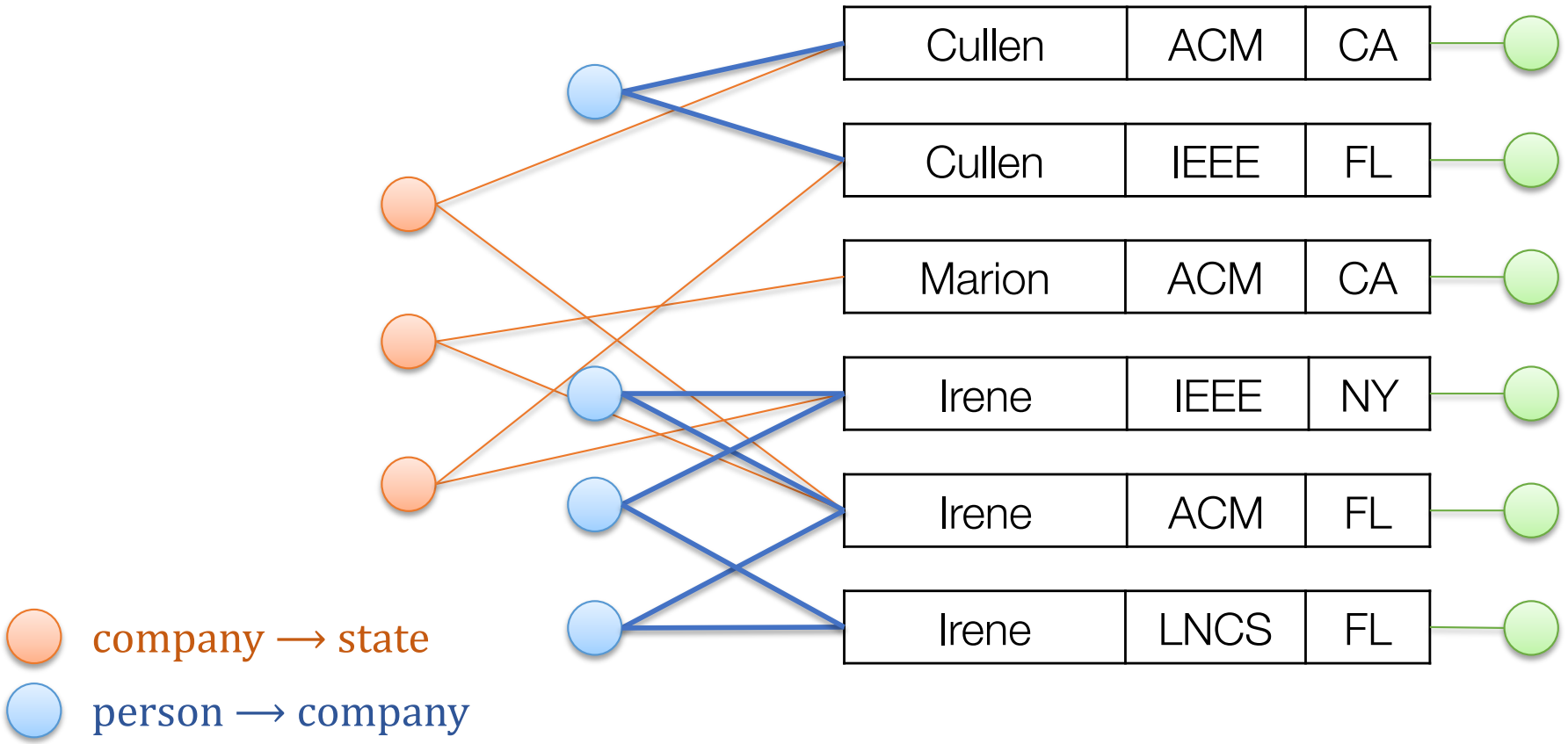
Realization

Probabilistic Noise Generator

Douglas	NYC	NY
Khan	Ghajar	Syria

Fundamental Open Problem in PUD

- We well understand the problem of finding an MPD (c-repair) under FD constraints [Livshits-K-Roy2018]
- What about MPD under **weak** FDs?



Concluding Remarks

Concluding Remarks

- Various concepts under the umbrella of PDB
  - [Source,Query,Target]-generative
- Part of the larger effort to democratize statistical modeling and complex inference
  - Hence, push for scale, expressiveness, precision, …
- Not the only reason to understand them!
  - *Do not be discouraged by not reaching standard DB volumes*
- Guiding principle to domain-independent semantics & error guarantees for a variety of DB ops
- Their computational complexity remarkably captures apparently unrelated problems
- Need more emphasis on fine-grained complexity of multiplicative approximation guarantees

Thank you!