Topics in Probabilistic and Statistical Databases

Lecture 3: Representation and Query Evaluation

Dan Suciu
University of Washington

Review of Representation

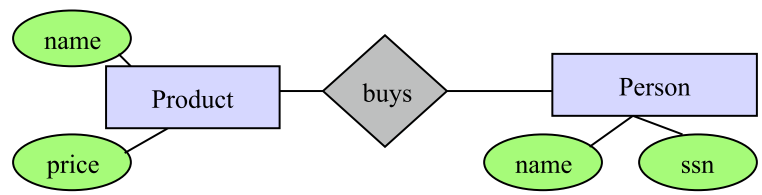
- Represent the possible worlds
- Represent the probability distribution

Review: Rule of Thumb #1

A ProbDB = an Incomplete DB + probabilities

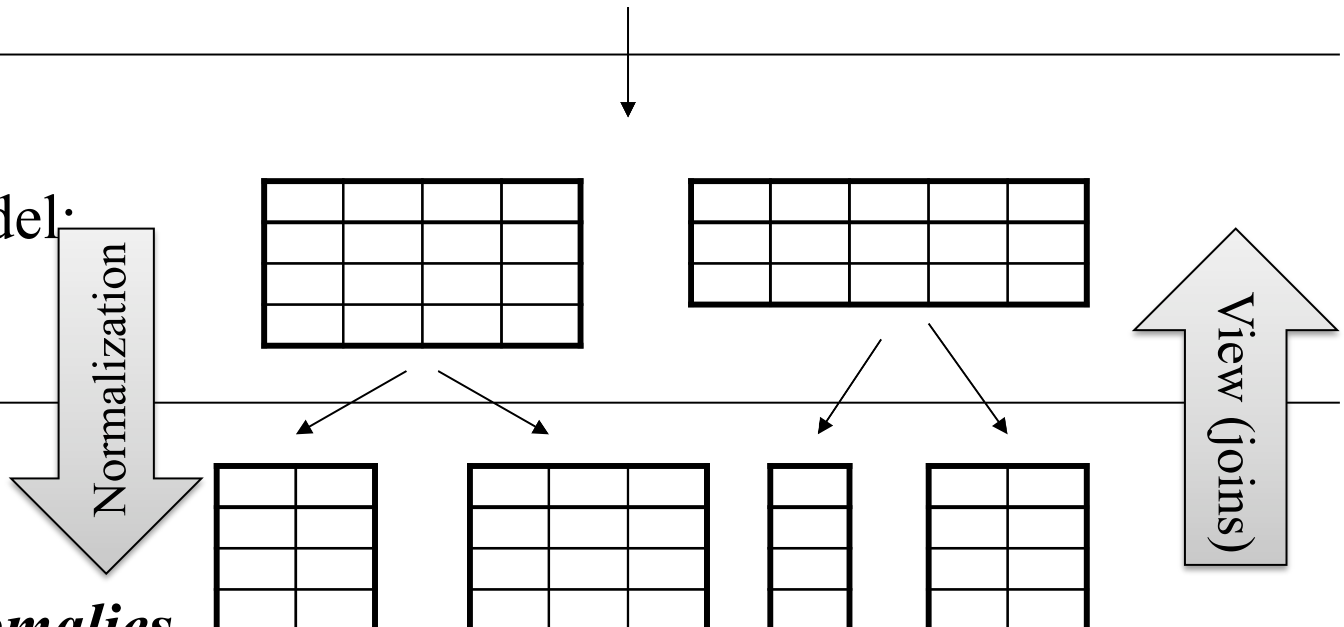
DB 101: Database Design

Conceptual Model:



Relational Model:

plus FD’s



Normalization:

Eliminates ***anomalies***

Review: Rule of Thumb #2

Probabilistic table =
Disjoint/independent tables + view (joins)

Discussion

Design

- How do we find the right decomposition ?
  - Normalization theory (n'th normal form…)
  - Probabilistic networks (same thing…)

Representation

- How to we recover the original table from the decomposed table(s) ?
  - At UW: we just use a SQL view
  - Elsewhere: equivalent formalism (e.g. U-relations)

[Antova'2008]

Representation: U-Relations

Two idea:

- Vertical decomposition
- Uniform lineage

Review: Vertical Decomposition

TS	Id	Type	Faction
a	1	Tank	Friend
b	3	Transport	Friend
c	3	Transport	Enemy
d	4	Transport	Enemy

TS	Id
a	1
b	3
c	3
d	4

TS	Type
a	Tank
b	Transport
c	Transport
d	Transport

TS	Faction
a	Friend
b	Friend
c	Enemy
d	Enemy

Review: Uniform Lineage

Saw

Witness	Car	X	V
Amy	Mazda	X1	1
Amy	Toyota	X1	2
Betty	Honda	X2	1

Drives

Person	Car	Y	W
Jimmy	Mazda	Y1	1
Jimmy	Toyota	Y1	2
Billy	Mazda	Y2	1
Billy	Honda	Y2	2

What is the lineage here ?

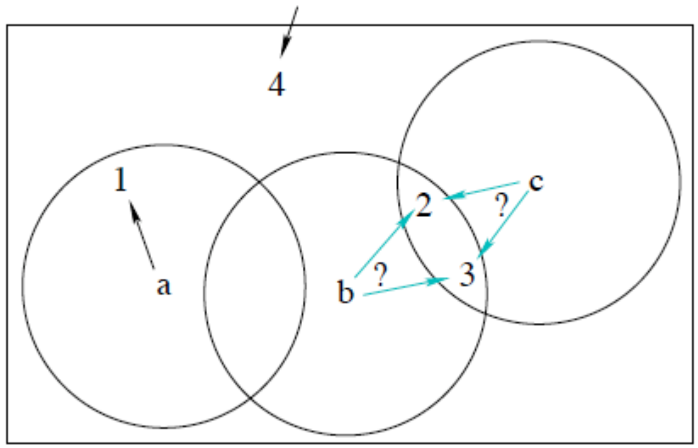
q(w,p) :- Saw(w,c),Drives(c,p)

Witness	Person	X	V	Y	W
Amy	Jimmy	X1	1	Y1	1
Amy	Jimmy	X1	2	Y1	2
Betty	Billy	X2	1	Y2	2

$X1=1 \land Y1=1 \lor$
$X1=2 \land Y1=2$

[Antova'2008]

U-Relations



W	Var	Rng
	x	1
	x	2
	y	1
	y	2
	z	1
	z	2

U_1	D	T_R	Id
		a	1
	$x \mapsto 1$	b	2
	$x \mapsto 2$	b	3
	$x \mapsto 1$	c	3
	$x \mapsto 2$	c	2
		d	4

U_2	D	T_R	Type
		a	Tank
		b	Transport
		c	Tank
	$y \mapsto 1$	d	Tank
	$y \mapsto 2$	d	Transport

U_3	D	T_R	Faction
		a	Friend
		b	Friend
		c	Enemy
	$z \mapsto 1$	d	Friend
	$z \mapsto 2$	d	Enemy

[Antova'2008]

Discussions

- Show the c-table R(TS, Id, Type Faction)
- Give the view expression that recovers R
- Is this a uniform lineage ?
- Do we need W ?

[Antova'2008]

Show the c-Table

D1	D2	D3	TS	Id	Type	Faction
			a	1	Tank	Friend
x→ 1			b	2	Transport	Friend
x→ 2			b	3	Transport	Friend
x→ 1			c	3	Tank	Enemy
x→ 2			c	2	Tank	Enemy
	y→ 1	z→ 1	d	4	Tank	Friend
	y→ 1	z→ 2	d	4	Tank	Enemy
	y→ 2	z→ 1	d	4	Transport	Friend
	y→ 2	z→ 2	d	4	Transport	Enemy

[Antova'2008]

Why We Need W

Need to store the probability distribution of the the variables separately

W	Var	Rng	Pr
	x	1	0.1
	x	2	0.9
	y	1	0.3
	y	2	0.7
	z	1	0.6
	z	2	0.4

[Antova'2008]

"Reduced" U-database

Definition A U-database is *reduced* if every tuple in every U-relation can appear in a possible world

U_1	D	T	A
	$c_1 \mapsto 1$	t_1	a_1
	$c_2 \mapsto 1$	t_2	a_2

U_2	D	T	B
	$c_1 \mapsto 1$	t_1	b_1
	$c_1 \mapsto 2$	t_1	b_2

This is not reduced
WHY ?

[Antova'2008]

More Examples

$$S = \pi_{\mathrm{Id}}(\sigma_{\mathrm{Type}='\mathrm{Tank}' \wedge \mathrm{Faction}='\mathrm{Enemy}'}(R))$$

U_4	D_1	D_2	T_S	Id
	$x \mapsto 1$		c	3
	$x \mapsto 2$		c	2
	$y \mapsto 1$	$z \mapsto 2$	d	4

$$(S\ s_1) \bowtie_{s_1.\mathrm{Id} \neq s_2.\mathrm{Id}} (S\ s_2)$$

U_5	D_1	D_2	D_3	T_{s_1}	T_{s_2}	Id_1	Id_2
	$x \mapsto 1$	$y \mapsto 1$	$z \mapsto 2$	c	d	3	4
	$x \mapsto 2$	$y \mapsto 1$	$z \mapsto 2$	c	d	2	4
	$y \mapsto 1$	$z \mapsto 2$	$x \mapsto 1$	d	c	4	3
	$y \mapsto 1$	$z \mapsto 2$	$x \mapsto 2$	d	c	4	2

Summary of U-Relations

U-relations = Vertical partition + uniform lineage

- Vertical partition: not necessarily disjoint attrs
  - Consistency check needed if multiple occurrences of A
- Reduced U-relations = semijoin reduction
- "Parsimonious query translation":
  - Select-project-join queries over possible worlds ➔ select-project-join queries over U-relations

The Design Problem

- How do we “normalize” a probabilistic database ?
- Combine:
  - MVD = multi-valued dependencies (read book..)
  - Probabilistic networks (next)

- NOTE: This is largely unexplored area

Conditional Independence

- Recall: $X \perp\!\!\!\perp Y \mid Z$ means:

$$\boxed{P(XY \mid Z) = P(X \mid Z)*P(Y \mid Z)}$$

- Equivalently:

$$\boxed{P(XYZ) = P(XZ) * P(YZ) / P(Z)}$$

Review: Rule of Thumb #3

Conditional independence =
MVD + some probability identities

Conditional Independence=MVD

- If $X \perp\!\!\!\perp Y \mid Z$ then $X \twoheadrightarrow Y$
- Stronger: $P(X,Y,Z) = P1(X,Z) \bowtie P2(Y,Z)$
  - This identity holds between *probabilistic tables*

Probabilistic Networks

- Many variables V1, V2, …, Vn
- Given joint distribution on their values
- Goal of a PN: to capture all conditional independences
- An edge (Vi, Vj) means, intuitively, that Vi, Vj are dependent
- Lack of an edge means they are independent

Quiz Time

- Three random variables V1, V2, V3
- Suppose their joint distribution is given by a function p(v1,v2,v3) = f(v1,v2)*g(v2,v3), where f, g are arbitrary positive functions

QUESTION What independence relation holds between V1, V2, V3 ?

Quiz Time

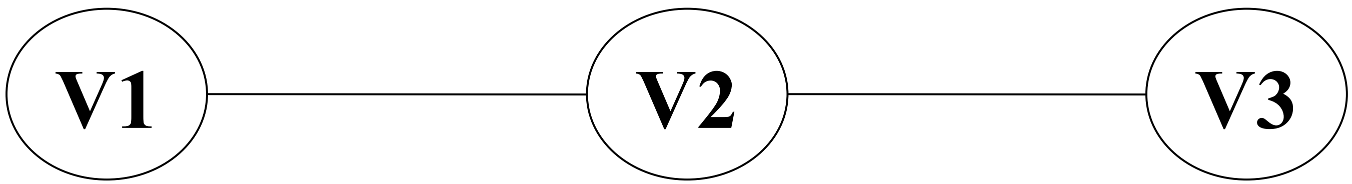
- Three random variables V1, V2, V3
- Suppose their joint distribution is given by a function p(v1,v2,v3) = f(v1,v2)*g(v2,v3), where f, g are arbitrary positive functions

QUESTION What independence relation holds between V1, V2, V3 ?

ANSWER $V1 \perp\!\!\!\perp V3 \mid V2$

(proof in class)

A Probabilistic Network

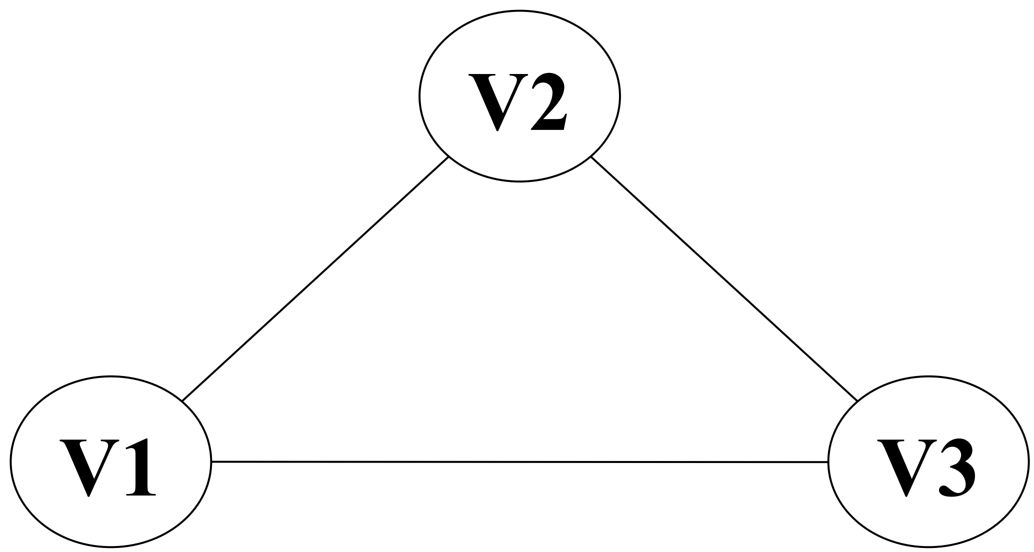


Quiz Time 2

- Now suppose
  p(v1,v2,v3) = f(v1,v2)*g(v2,v3)*h(v1,v3)

QUESTION What independence relation
  holds between V1, V2, V3 ?

Another Probabilistic Network



Answer: NONE !

[Cowell'99]

Hierarchical Decomposition

- $P(V_1, \ldots, V_n)$ is a *Hierarchical Distribution* if there exists functions $f_1, \ldots, f_k$, s.t.:
  - $P = f_1 * f_2 * \ldots * f_k$
  - $f_i(\mathcal{V}_{Ci})$ depends on a subset $\mathcal{V}_{ci} \subseteq \{V_1, \ldots, V_n\}$
- Also called: *Probabilistic Graphical Model*
- $f_1, \ldots, f_k$ (or the set $\mathcal{V}_{ci}$) are called *factors*

[Cowell'99]

Probabilistic Network

- Define a graph G = (V, E) s.t.
  - $V = \{V_1, \ldots, V_n\}$
  - $E = \{(V_i, V_j) \mid \text{exists } \mathcal{V}_c \text{ s.t. } V_i \in \mathcal{V}_c, V_j \in \mathcal{V}_c\}$
- A *clique* is $C \subseteq V$ s.t. any two nodes in C are connected
- A *separator* is $S = C \cap C'$, where C, C'=cliques

[Cowell'99]

Probabilistic Network

- Let C1, C2, … be all the cliques, and S1, S2, … all separators
- Then:

$$P(V) = P(\mathcal{V}_{C1})*P(\mathcal{V}_{C2})*\ldots/(P(\mathcal{V}_{S1})*P(\mathcal{V}_{S2})*\ldots.)$$

Research question: how does decompose P into D/I tables + views ?

Query Evaluation on D/I Databases

Problem Statement

- Given:
  - A disjoint/independent probdb PDB
  - A Boolean conjunctive query Q
- Compute the probability Q(PDB)

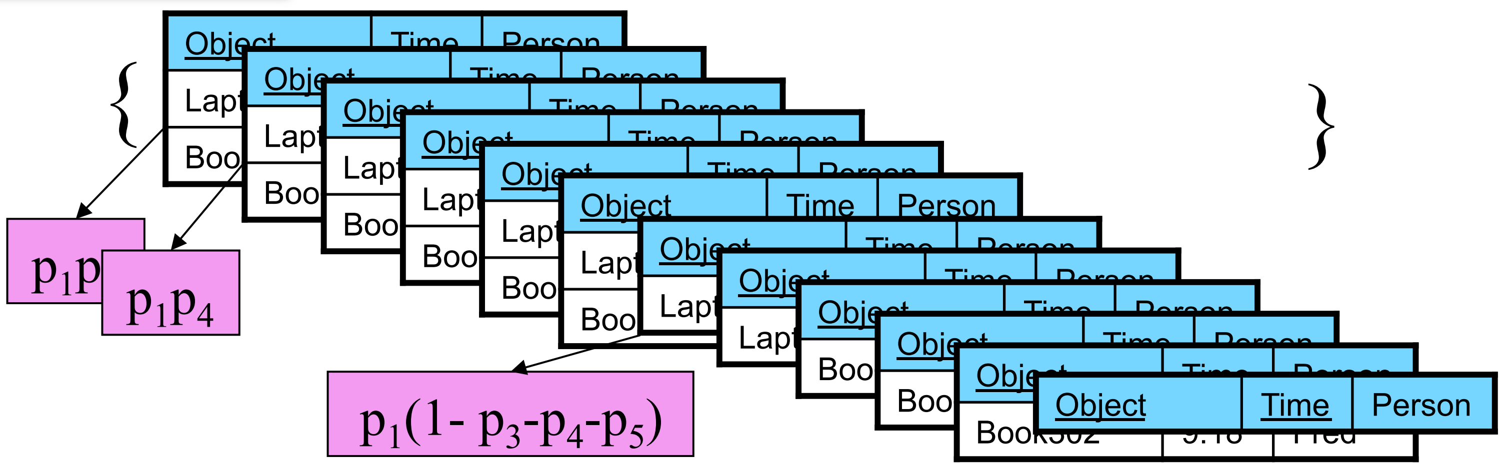
Why does it suffice to restrict to D/I databases ?

Review: D/I Databases

S =

<u>Object</u>	<u>Time</u>	Person	P
LaptopX77	9:07	John	p_1
LaptopX77	9:07	Jim	p_2
Book302	9:18	Mary	p_3
Book302	9:18	John	p_4
Book302	9:18	Fred	p_5

Rep(S) =



Review Query Semantics

Semantics 1: Possible Sets of Answers

A probability distributions on *sets of tuples*

$$8\ A.\ \Pr(Q = A) = \sum_{I\ 2\ \mathbf{Inst}.\ Q(I) = A} \Pr(I)$$

Semantics 2: Possible Tuples

A probability function on *tuples*

$$8\ t.\ \Pr(t\ 2\ Q) = \sum_{I\ 2\ \mathbf{Inst}.\ t2\ Q(I)} \Pr(I)$$

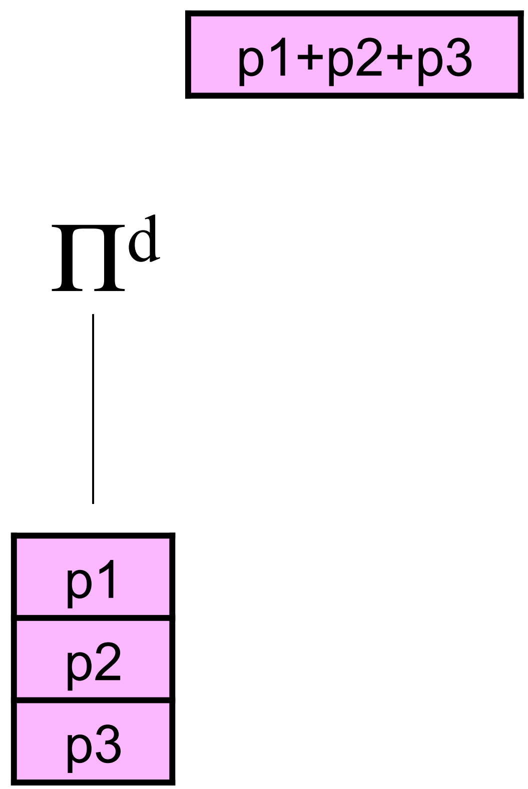
Extensional Operators

Object	Person	Location	P
Laptop77	John	L45	p1
	Jim	L45	p2
	Jim	L66	p3
Book302	Mary	L66	p4
	Mary	L45	p5
	Jim	L66	p6
	John	L45	p7
	Fred	L45	p8

$q(z)$:- $HasObject^p($**Book302**$, y, z)$

Location	P
L66	p4+p6
L45	p5+p7+p8

Disjoint Project



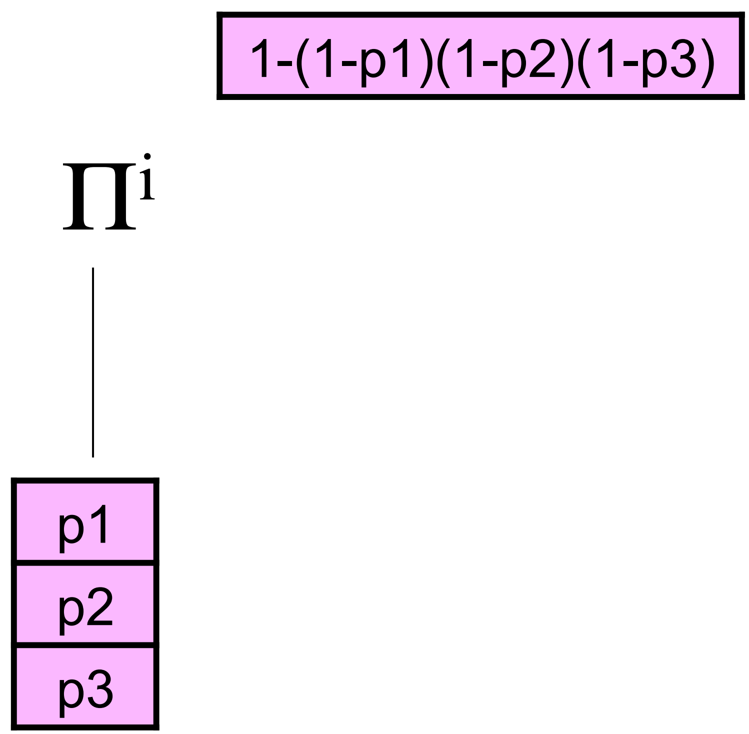
Extensional Operators

Object	Person	Location	P
Laptop77	John	L45	p1
	Jim	L45	p2
	Jim	L66	p3
Book302	Mary	L66	p4
	Mary	L45	p5
	Jim	L66	p6
	John	L45	p7
	Fred	L45	p8

$q(y,z)$:- $HasObject^p(\underline{\mathbf{x}},y,z)$

Person	Location	P
Jim	L66	1-(1-p3)(1-p6)
John	L45	1-(1-p1)(1-p7)
. . .		

Independent Project



$q(y)$:- $\text{Movie}^p(\underline{x},\underline{y})$, $\text{Review}^p(\underline{x},\underline{z})$, $z>3$

Example

Movie

id	year	P
m42	1995	p1
m99	2002	p2
m76	2002	p3

Review

mid	rating	P
m42	7	q1
m42	4	q2
m42	9	q3
m99	7	q4
m99	5	q5
m76	6	q6

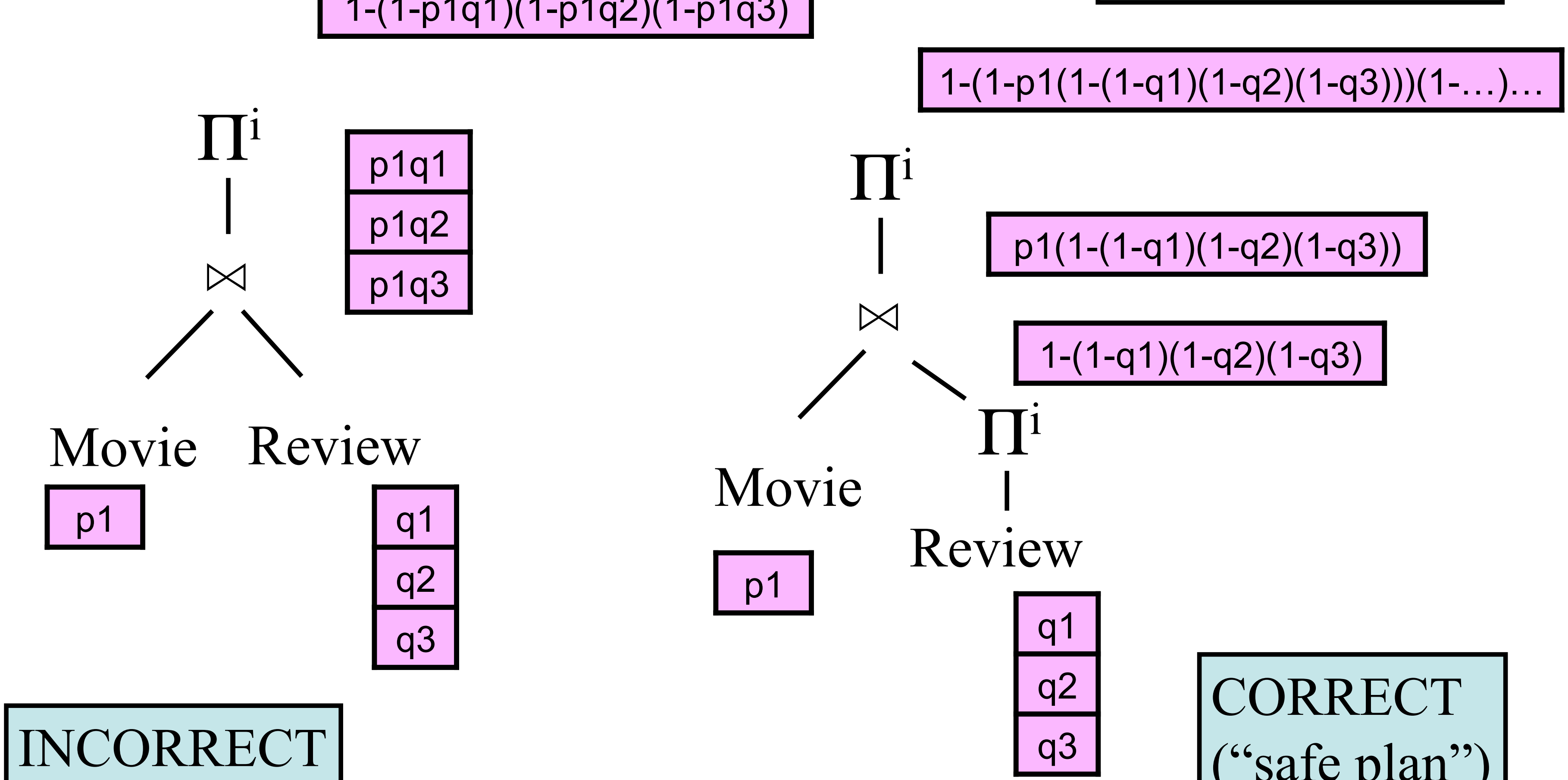
Answer

year	P
1995	$p1 \times (1 - (1 - q1)\times(1 - q2)\times(1 - q3))$
2002	$1 - (1 - p2 \times (1 - (1 - q4)\times(1 - q5))) \times (1 - p3 \times q6)$

$q(y)$:- $Movie^p(\underline{\mathbf{x}},\underline{\mathbf{y}})$, $Review^p(\underline{\mathbf{x}},\underline{\mathbf{z}})$

q(1995)

Answer depends on query plan !



Safe Plans are Efficient

- Very efficient: run almost as fast as regular queries
- Require only simple modifications of the relational operators
- Or can be translated back into SQL and sent to any RDBMS

Can we always generate a safe plan ?

A Hard Query

R^p

A	B	P
a	x1	p1
a	x2	p2

S

B	C
x1	y1
x1	y2
x2	y1

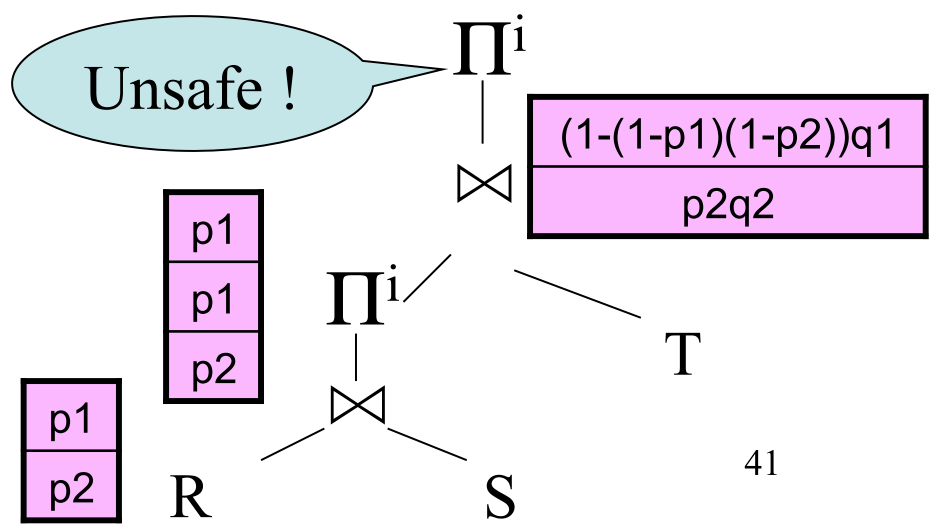
T^p

C	D	P
y1	c	q1
y2	c	q2

$h(u,v)$:- $R^p(\underline{\mathbf{u}},\underline{\mathbf{x}}), S(\underline{\mathbf{x}},\underline{\mathbf{y}}), T^p(\underline{\mathbf{y}},\underline{\mathbf{v}})$

$h(a,c)$

There is no safe plan !



Independent Queries

Let q1, q2 be two boolean queries

<u>Definition</u> q1, q2 are "independent" if P(q1, q2) = P(q1) P(q2)

Also: $P(q1 \lor q2) = 1 - (1 - P(q1))(1 - P(q2))$

Quiz: which are independent ?

q1	q2	Indep.?
$Movie^p(\underline{\mathbf{m41}},\underline{\mathbf{y}})$	$Review^p(\underline{\mathbf{m41}}, \underline{\mathbf{z}})$	
$Movie^p(\underline{\mathbf{m42}},\underline{\mathbf{y}}),Review^p(\underline{\mathbf{m42}},\underline{\mathbf{z}})$	$Movie^p(\underline{\mathbf{m77}},\underline{\mathbf{y}}),Review^p(\underline{\mathbf{m77}},\underline{\mathbf{z}})$	
$Movie^p(\underline{\mathbf{m42}},\underline{\mathbf{y}}),Review^p(\underline{\mathbf{m42}},\underline{\mathbf{z}})$	$Movie^p(\underline{\mathbf{m42}}, \underline{\mathbf{1995}})$	
$Movie^p(\underline{\mathbf{m42}},\underline{\mathbf{y}}),Review^p(\underline{\mathbf{m42}},\underline{\mathbf{7}})$	$Movie^p(\underline{\mathbf{m42}},\underline{\mathbf{y}}),Review^p(\underline{\mathbf{m42}},\underline{\mathbf{4}})$	
$R^p(\underline{\mathbf{x}},\underline{\mathbf{y}},\underline{\mathbf{z}},\underline{\mathbf{z}},\underline{\mathbf{u}}), R^p(\underline{\mathbf{x}},\underline{\mathbf{x}},\underline{\mathbf{x}},\underline{\mathbf{y}},\underline{\mathbf{y}})$	$R^p(\underline{\mathbf{a}},\underline{\mathbf{a}},\underline{\mathbf{b}},\underline{\mathbf{b}},\underline{\mathbf{c}})$	

Answers

q1	q2	Indep.?
$Movie^p$(**<u>m41</u>**,<u>y</u>)	$Review^p$(**<u>m41</u>**, <u>z</u>)	YES
$Movie^p$(**<u>m42</u>**,<u>y</u>),$Review^p$(**<u>m42</u>**,<u>z</u>)	$Movie^p$(**<u>m77</u>**,<u>y</u>),$Review^p$(**<u>m77</u>**,<u>z</u>)	YES
$Movie^p$(**<u>m42</u>**,<u>y</u>),$Review^p$(**<u>m42</u>**,<u>z</u>)	$Movie^p$(**<u>m42</u>**, **<u>1995</u>**)	NO
$Movie^p$(**<u>m42</u>**,<u>y</u>),$Review^p$(**<u>m42</u>**,<u>7</u>)	$Movie^p$(**<u>m42</u>**,<u>y</u>),$Review^p$(**<u>m42</u>**,<u>4</u>)	NO
R^p(<u>x</u>,<u>y</u>,<u>z</u>,<u>z</u>,<u>u</u>), R^p(<u>x</u>,<u>x</u>,<u>x</u>,<u>y</u>,<u>y</u>)	R^p(**<u>a</u>**,**<u>a</u>**,**<u>b</u>**,**<u>b</u>**,**<u>c</u>**)	YES

<u>Prop</u> If no two subgoals unify then q1,q2 are independent

Detour: Independent Queries

- Let q= a Boolean conjunctive query
- A critical tuple is a tuple t s.t.
  - There exists an instance I s.t.
    - q(I) is true
    - q(I-{t}) is false
  - (in other words, $I \vDash q$ and $I-\{t\} \nvDash q$)
- Denote crit(q) = the set of critical tuples

Detour: Independent Queries

Note: *necessary* but not *sufficient* condition

Theorem Queries q1, q2 are independent iff crit(q1) ∩ crit(q2) = ∅

q1 :- $Movie^p$(**m42**,**y**),$Review^p$(**m42**,**z**)

Crit(q1) = ?

q2 :- $Movie^p$(**m77**,**y**),$Review^p$(**m77**,**z**)

Crit(q2) = ?

Detour: Independent Queries

Note: *necessary* but not *sufficient* condition

q1 :– R(x, y, z, z, u),R(x, x, x, y, y).

q2 :– R(a, a, b, b, c),

Theorem Independence is Π^p_2 complete [Miklau&S'04]
Reducible to query containment [Machanavajjhala&Gehrke'06]

Disjoint Queries

Let q1, q2 be two boolean queries

Definition q1, q2 are "disjoint" if P(q1, q2) = 0

Iff q1, q2 depend on two disjoint tuples t1, t2

Quiz: which are disjoint ?

q1	q2	?
HasObject^p(‘**book**’, ‘**9**’, ‘Mary’, x)	HasObject^p(‘**book**’, ‘**9**’, ‘Jim’, x)	
HasObject^p(‘**book**’, **t**, ‘Mary’, x)	HasObject^p(‘**book**’, **t**, ‘Jim’, x)	
HasObject^p(‘**book**’, ‘**9**’, u, x)	HasObject^p(‘**book**’, ‘**9**’, v, x)	

Answers

q1	q2	?
HasObject^p(**<u>'book'</u>**, **<u>'9'</u>**, 'Mary', x)	HasObject^p(**<u>'book'</u>**, **<u>'9'</u>**, 'Jim', x)	Y
HasObject^p(**<u>'book'</u>**, **<u>t</u>**, 'Mary', x)	HasObject^p(**<u>'book'</u>**, **<u>t</u>**, 'Jim', x)	N
HasObject^p(**<u>'book'</u>**, **<u>'9'</u>**, u, x)	HasObject^p(**<u>'book'</u>**, **<u>'9'</u>**, v, x)	N

<u>Proposition</u> q1, q2 are "disjoint" if they contain subgoals g1, g2:

Have the same values for the key attributes

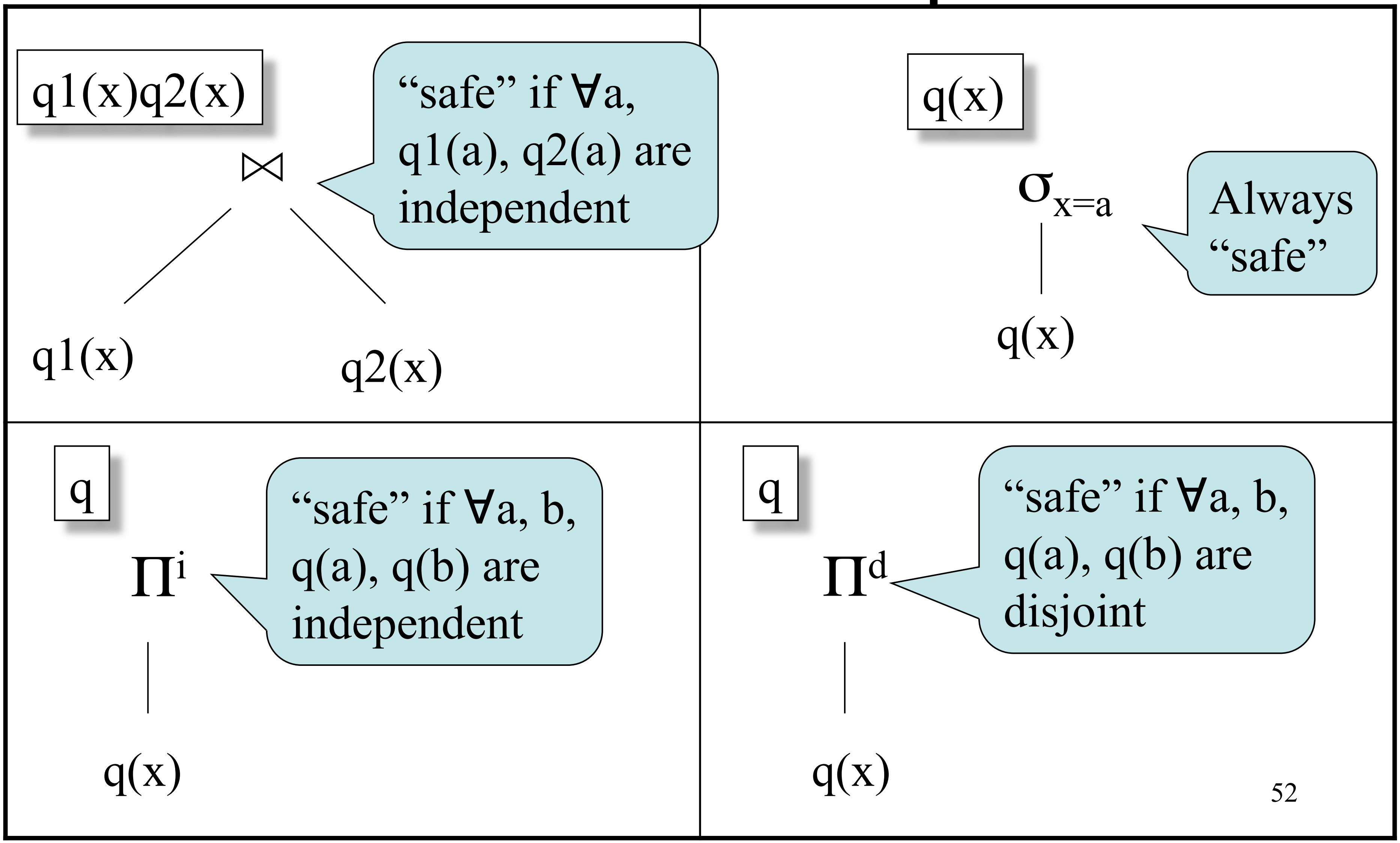
these values are constants

have at least one different constant in the non-key attributes

Definition of Safe Operators

- Safe join = left and right are indep.
- Safe independent project = duplicate tuples are independent
- Safe disjoint project = duplicate tuples are disjoint
- Safe select = any select is safe ☺

Definition of Safe Operators

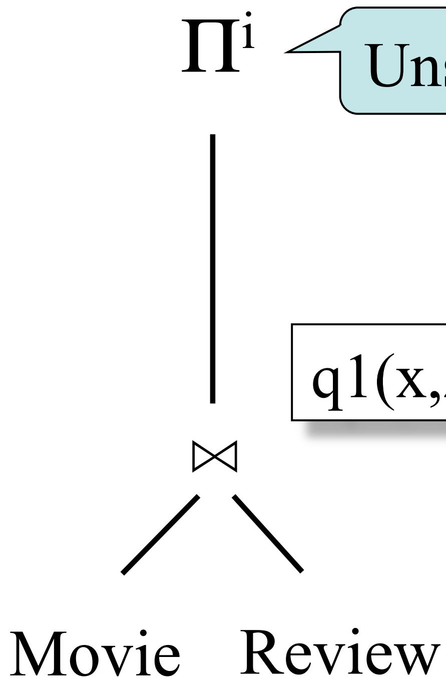


$q(y^c)$:- $Movie^p(\underline{\mathbf{x}},y^c)$, $Review^p(\underline{\mathbf{x}},\underline{\mathbf{z}})$

y^c "is a constant"

Example 1

$q1$:- $Movie(\underline{\mathbf{x}},y^c)$, $Review(\underline{\mathbf{x}},\underline{\mathbf{z}})$



Unsafe

Because these are dependent:

$q1(m42,7)=Movie(m42,y^c),Review(m42,7)$

$q1(m42,4)=Movie(m42,y^c),Review(m42,7)$

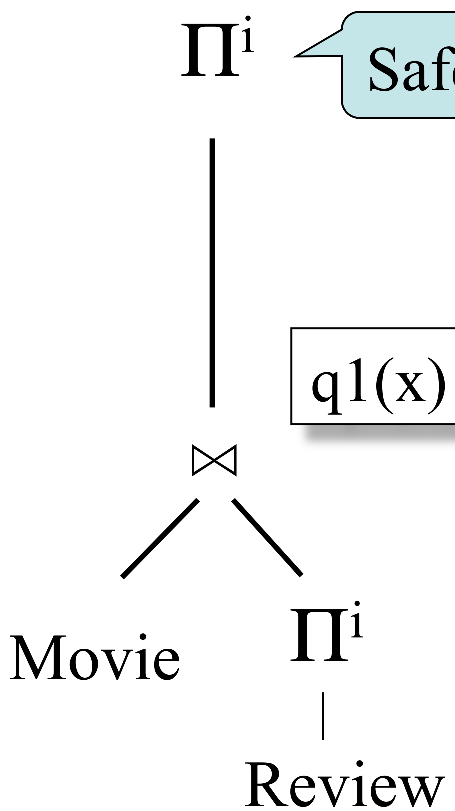
$q1(x,z)$:- $Movie(\underline{\mathbf{x}},y^c)$, $Review(\underline{\mathbf{x}},\underline{\mathbf{z}})$

$q(y^c)$:- $\text{Movie}^p(\underline{\mathbf{x}},y^c)$, $\text{Review}^p(\underline{\mathbf{x}},\underline{\mathbf{z}})$

y^c "is a constant"

Example 2

$q1$:- $\text{Movie}(\underline{\mathbf{x}},y^c)$, $\text{Review}(\underline{\mathbf{x}},\underline{\mathbf{z}})$



Safe !

Now these are independent !

$q1(m42)$ = $\text{Movie}(m42,y^c)$, $\text{Review}(m42,z)$

$q1(m77)$ = $\text{Movie}(m77,y^c)$, $\text{Review}(m77,z)$

$q1(x)$:- $\text{Movie}(\underline{\mathbf{x}},y^c)$, $\text{Review}(\underline{\mathbf{x}},\underline{\mathbf{z}})$

[Valiant'79]

Complexity Class #P

<u>Definition</u> #P is the class of functions f(x) for which there exists a PTIME non-deterministic Turing machine M s.t. f(x) = number of accepting computations of M on input x

Examples:

SAT = "given formula Φ, is Φ satisfiable ?"
= NP-complete

#SAT = "given formula Φ, count # of satisfying assignments"
= #P-complete

[Valiant'79] [Provan&Ball'83]

Examples

Class	Example	SAT	#SAT
3CNF	$(X \vee Y \vee Z) \wedge (\neg X \vee U \vee W) \ldots$	NP	#P
2CNF	$(X \vee Y) \wedge (\neg X \vee U) \ldots$	PTIME	#P
Positive, partitioned 2CNF	$(X1 \vee Y1) \wedge (X1 \vee Y4) \wedge (X2 \vee Y1) \wedge (X3 \vee Y1) \ldots$	PTIME	#P
Positive, partitioned 2DNF	$(X1 \wedge Y1) \vee (X1 \wedge Y4) \vee (X2 \wedge Y1) \vee (X3 \wedge Y1) \ldots$	PTIME	#P

Here NP, #P means "NP-complete, #P-complete"

See also [Graedel et al. 98]

#P-Hard Queries

hd1 :- R^p(**x**),S(**x**,**y**),T^p(**y**)

Theorem The query hd1 is #P-hard

Proof: Reduction from partitioned, positive 2DNF

E.g. $\varnothing$ = x1 y1 $\vee$ x2 y1 $\vee$ x1 y2 $\vee$ x3 y2 reduces to

R^p

A	P
x1	0.5
x2	0.5
x3	0.5

S

A	B
x1	y1
x2	y1
x1	y2
x3	y2

T^p

B	P
y1	0.5
y2	0.5

$\#\varnothing = P(hd1) * 2^n$

Where We Are

- We have seen a query that can be evaluated with a safe plan
  – Very efficient

- We have seen a query whose data complexity is #P hard

- What is the general picture ?

PTIME Queries

R(<u>**x, y**</u>), S(<u>**x, z**</u>)

R(<u>**x**</u>, y), S(<u>**y**</u>), T(<u>**'a'**</u>, y)

R(<u>**x**</u>), S(<u>**x, y**</u>), T(<u>**y**</u>), U(<u>**u**</u>, y), W(<u>**'a'**</u>, u)

. . .

#P-Hard Queries

hd1 = R(<u>**x**</u>), S(<u>**x, y**</u>), T(<u>**y**</u>)

hd2 = R(<u>**x**</u>,y), S(<u>**y**</u>)

hd3 = R(<u>**x**</u>,y), S(x,<u>**y**</u>)

. . .

Will discuss next how to decide their complexity and how evaluate PTIME queries

Dichotomy for a Language L

- Fix a query language L
- The dichotomy property is:
  - Every query in L is either in PTIME or #P-hard
- Note that this does not follow from general principles
  - It may be false for some languages L

Dichotomy Property

LANG: CQ = conjunctive queries
CQ^1 = conjunctive queries without self-joins

<u>Theorem</u> The dichotomy property holds for:

CQ^1 and independent dbs.

CQ^1 and disjoint/independent dbs.

CQ and independent dbs.

We'll start these today and continue next lecture

Hierarchical Queries

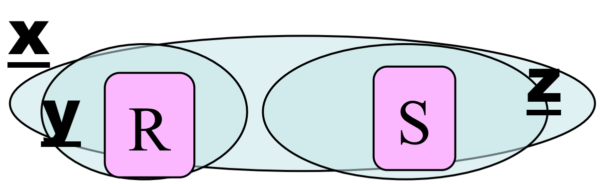
sg(x) = set of subgoals containing the variable x in a key position

Definition A query q is *hierarchical* if forall x, y:

sg(x) ⊆ sg(y) or sg(y) ⊆ sg(x) or sg(x) ∩ sg(y) = ∅

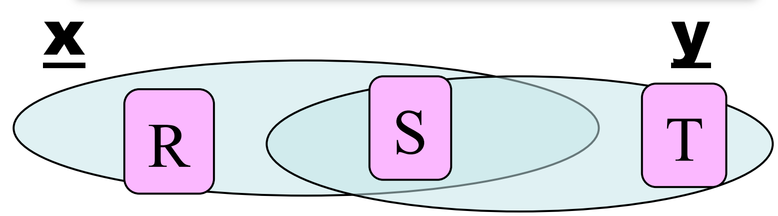
Hierarchical

q = R(**x, y**), S(**x, z**)



Non-hierarchical

h1 = R(**x**), S(**x, y**), T(**y**)



[Dalvi&S'2004]

Case 1: CQ^1 + Independent

Note that in this case:

- CQ^1 (conjunctive queries, no self-joins):
  - $R(\underline{\mathbf{x}},\underline{\mathbf{y}}), S(\underline{\mathbf{y}},\underline{\mathbf{z}})$ OK
  - $R(\underline{\mathbf{x}},\underline{\mathbf{y}}), R(\underline{\mathbf{y}},\underline{\mathbf{z}})$ Not OK
- Independent tuples only:
  - $R(\underline{\mathbf{x}},\underline{\mathbf{y}})$ OK
  - $S(\underline{\mathbf{y}},z)$ Not OK

[Dalvi&S'2004]

CQ^1 + Independent

Theorem Forall $q \in CQ^1$:

q is hierarchical, has a safe plan, and is in PTIME,

OR

q is not hierarchical and is #P-hard

The PTIME Queries

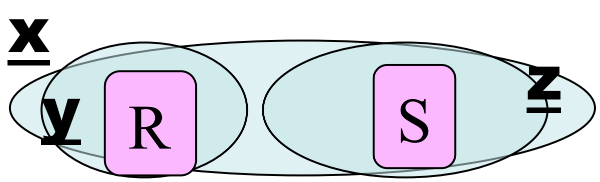
Algorithm: convert a Hierarchy to a Safe Plan

Root variable u ➔ Π^i_{-u}

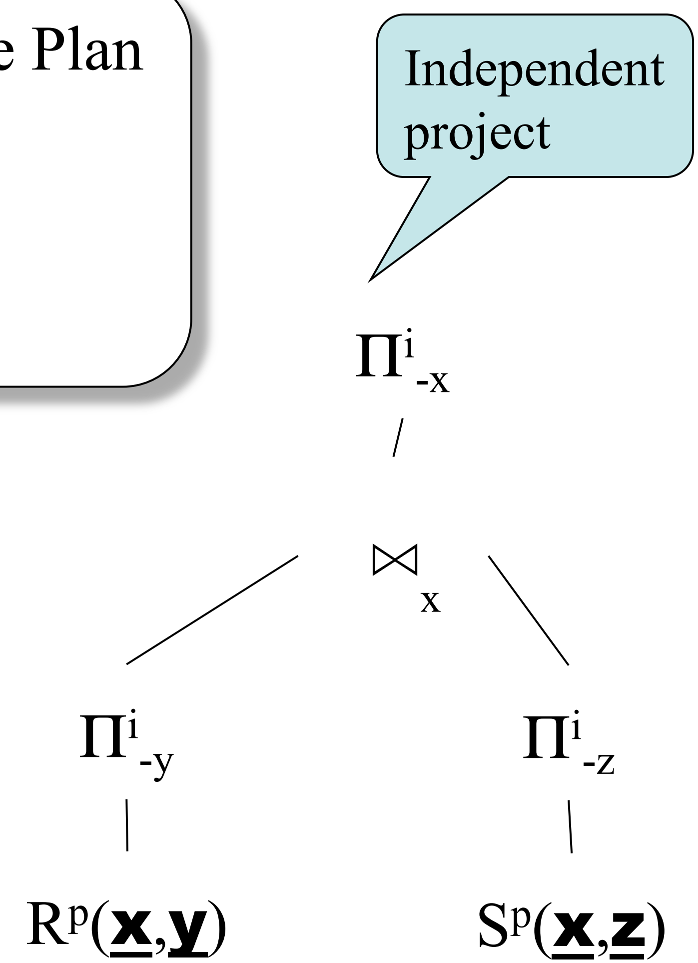
Connected components ➔ Join

Single subgoal ➔ Leaf node

q = R(**x**, **y**), S(**x**, **z**)



➔



$P(q) =$
$1 - (1-p_1(1-(1-q_1)(1-q_2))) * (1-p_2(1-(1-q_3)(1-q_4)(1-q_5)))$

Π_{-x}

A	P
a_1	$p_1(1-(1-q_1)(1-q_2))$
a_2	$p_2(1-(1-q_3)(1-q_4)(1-q_5))$

q =

R(**x**, **y**), S(**x**, **z**)

$\bowtie_x$

A	P
a_1	$1-(1-q_1)(1-q_2)$
a_2	$1-(1-q_3)(1-q_4)(1-q_5)$

Π_{-y} Π_{-z}

R^p(**x**,**y**)

A	**B**	P
a_1	b_1	p_1
a_2	b_2	p_2

S^p(**x**,**z**)

A	**C**	P
a_1	c_1	q_1
a_1	c_2	q_2
a_2	c_3	q_3
a_2	c_4	q_4
a_2	c_5	q_5

[D&S'2004]

The #P-Hard Queries

Are precisely the non-hierarchical queries. Example:

hd1 :- R(<u>**x**</u>), S(<u>**x, y**</u>), T(<u>**y**</u>)

More general:

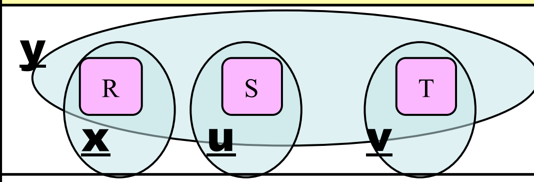
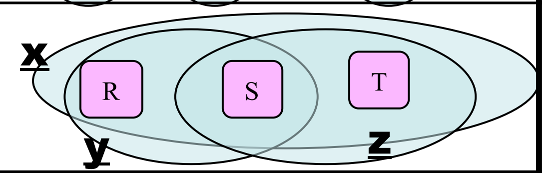
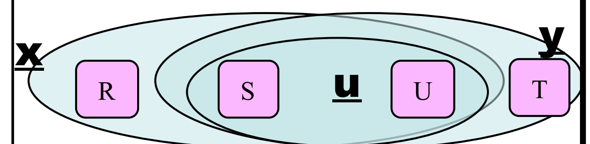
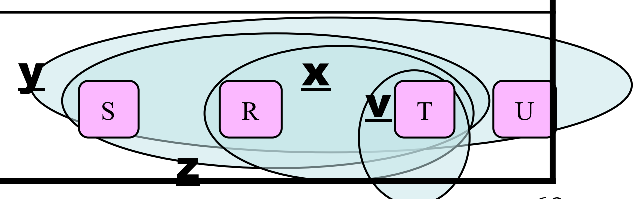
q :- …, R(<u>**x**</u>, …), S(<u>**x, y**</u>, …), T(<u>**y**</u>, …) , …

<u>**Theorem**</u> Testing if q is PTIME or #P-hard is in AC^0

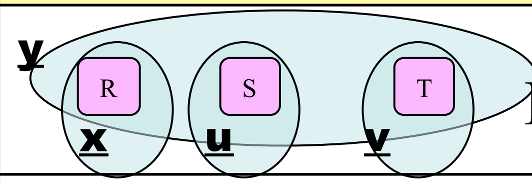
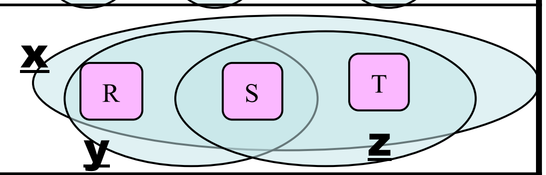
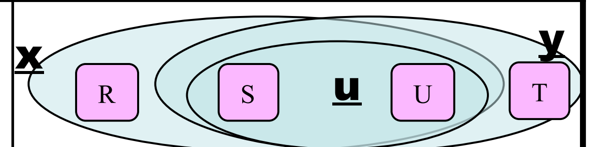
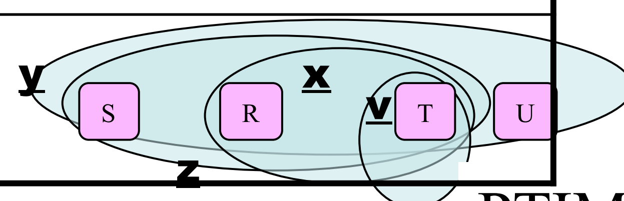
Quiz: What is their complexity ?

q	PTIME or #P ?
R(**x**,**y**),S(**y**,**a**,**u**),T(**y**,**y**,**v**)	
R(**x**,**y**), S(**x**,**y**,**z**), T(**x**,**z**)	
R(**x**,**a**),S(**y**,**u**,**x**),T(**u**,**y**),U(**x**,**y**)	
R(**x**,**y**,**z**),S(**z**,**u**,**y**),T(**y**,**v**,**z**,**x**),U(**y**)	

Hint…

q	PTIME or #P ?
R(**x**,**y**),S(**y**,**a**,**u**),T(**y**,**y**,**v**)	
R(**x**,**y**), S(**x**,**y**,**z**), T(**x**,**z**)	
R(**x**,**a**),S(**y**,**u**,**x**),T(**u**,**y**),U(**x**,**y**)	
R(**x**,**y**,**z**),S(**z**,**u**,**y**),T(**y**,**v**,**z**,**x**),U(**y**)	

…Answer

q	PTIME or #P ?
R(**x**,**y**),S(**y**,**a**,**u**),T(**y**,**y**,**v**)	 PTIME
R(**x**,**y**), S(**x**,**y**,**z**), T(**x**,**z**)	 #P
R(**x**,**a**),S(**y**,**u**,**x**),T(**u**,**y**),U(**x**,**y**)	 #P
R(**x**,**y**,**z**),S(**z**,**u**,**y**),T(**y**,**v**,**z**,**x**),U(**y**)	 PTIME

Summary

- We have discussed only the simplest case: CQ w/o self-joins, on independent dbs
- Next time:
  - Add FDs at the representation level
  - Extend to independent/disjoint dbs
  - Extend to arbitrary CQs (with self-joins)