



Einsatz und Realisierung von Datenbanksystemen

ERDB Übungsleitung

Maximilian {Bandle, Schüle}, Josef Schmeißer

i3erdb@in.tum.de

Folien erstellt von Maximilian Bandle & Alexander Beischl



Organisatorisches

Disclaimer

Die Folien werden von der Übungsleitung allen Tutoren zur Verfügung gestellt.

Sollte es Unstimmigkeiten zu den Vorlesungsfolien von Prof. Kemper geben, so sind die Folien aus der Vorlesung ausschlaggebend.

Falls Ihr einen Fehler oder eine Unstimmigkeit findet, schreibt an i3erdb@in.tum.de mit Angabe der Foliennummer.

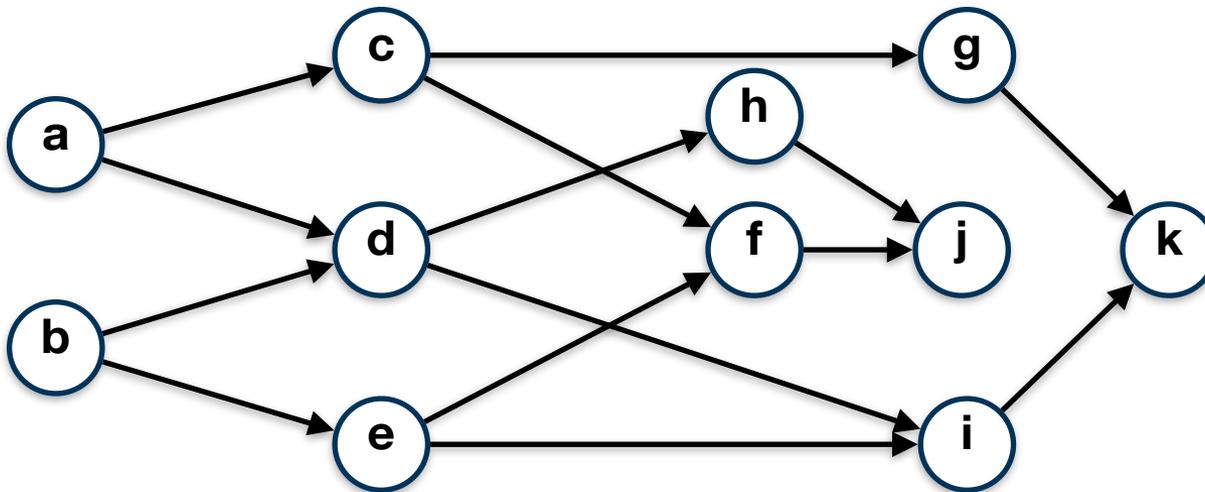
Aufgabe 1

Definieren Sie das Prädikat $sg(X,Y)$ das für “same generation” steht. Zwei Personen gehören zur selben Generation, wenn Sie mindestens je ein Elternteil haben, das derselben Generation angehört.

Verwenden Sie beispielsweise die folgende Ausprägung einer ElternKind Relation. Das erste Element ist hier das Kind, das Zweite ein Elternteil.

- Definieren Sie das Prädikat in Datalog.
- Demonstrieren Sie die naive Ausführung des Prädikats.
- Erläutern Sie das Vorgehen bei der seminaiven Auswertung.

```
parent(c,a).  
parent(d,a).  
parent(d,b).  
parent(e,b).  
parent(f,c).  
parent(g,c).  
parent(h,d).  
parent(i,d).  
parent(i,e).  
parent(f,e).  
parent(j,f).  
parent(j,h).  
parent(k,g).  
parent(k,i).
```





Aufgabe 5

Naive Auswertung

$S := \{\};$

repeat

$S' := S;$

$S := \Pi_{X,Y} (P(Z, X) \bowtie_{X=Y} P(Z, Y));$

$S := S(X, Y) \cup \Pi_{X,Y} (P(X, Z) \bowtie_{X=Y} P(Y, Z));$

$S := S(X, Y) \cup \Pi_{X,Y} (P(X, U) \bowtie (S'(U, V) \bowtie P(Y, V)));$

until $S' = S$

output $S;$



Aufgabe 5

Semi-naive Auswertung

```
S := {}; ΔS := {};  
ΔS := ΠX,Y (P(Z, X) ⋈X=Y P(Z, Y));  
ΔS := ΔS(X, Y) ∪ ΠX,Y (P(X, Z) ⋈X=Y P(Y, Z));  
ΔS := ΔS(X, Y) ∪ ΠX,Y (P(X, U) ⋈ (S(U, V) ⋈ P(Y, V)));  
S := ΔS;  
repeat  
  ΔS' := ΔS;  
  ΔS := ΠX,Y (P(Z, X) ⋈X=Y ΔP(Z, Y)) !erste und*!  
    ∪ ΠX,Y (ΔP(Z, X) ⋈X=Y P(Z, Y))  
    ∪ ΠX,Y (P(X, Z) ⋈X=Y ΔP(Y, Z)) !zweite Regel*!  
    ∪ ΠX,Y (ΔP(X, Z) ⋈X=Y P(Y, Z)); !liefern ∅*!  
  ΔS := ΔS  
    ∪ ΠX,Y (ΔP(X, U) ⋈ (S(U, V) ⋈ P(Y, V))) !liefert ∅*!  
    ∪ ΠX,Y (P(X, U) ⋈ (ΔS'(U, V) ⋈ P(Y, V))) !kann ≠ ∅ sein*!  
    ∪ ΠX,Y (P(X, U) ⋈ (S(U, V) ⋈ ΔP(Y, V))); !liefert ∅*!  
  ΔS := ΔS - S; !entferne Tupel, die schon vorhanden waren*!  
  S := S ∪ ΔS;  
until ΔS = ∅
```



Deduktive Datenbanken

Naive Auswertung der Rekursion

parent \leq (K1,K11), (K1,K12), (K11,K111), (K11,K112), (K12,K121),
(K12,K122)

verwandte(Vor, Nach) :- parent(Vor, Nach)

verwandte(Vor, Nach) :- verwandte(Vor, Mitte), parent(Mitte, Nach)

Schritt 0: (K1,K11), (K1,K12), (K11,K111), (K11,K112), (K12,K121),
(K12,K122)

Schritt 1: (K1,K11), (K1,K12), (K11,K111), (K11,K112), (K12,K121),
(K12,K122), (K1, K111), (K1, K112), (K1, K121), (K1, K122)

Schritt 2: (K1,K11), (K1,K12), (K11,K111), (K11,K112), (K12,K121),
(K12,K122), (K1, K111), (K1, K112), (K1, K121), (K1, K122)



Deduktive Datenbanken

Semi-Naive Auswertung der Rekursion

parent \leq (K1,K11), (K1,K12), (K11,K111), (K11,K112), (K12,K121),
(K12,K122)

verwandte(Vor, Nach) :- parent(Vor, Nach)

verwandte(Vor, Nach) :- verwandte(Vor, Mitte), parent(Mitte, Nach)

Schritt 0: (K1,K11), (K1,K12), (K11,K111), (K11,K112), (K12,K121),
(K12,K122)

Deduktive Datenbanken

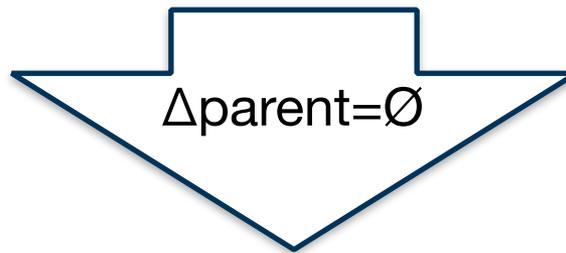
Semi-Naive Auswertung der Rekursion

Nur weiteres Auswerten der neu hinzugefügten Knoten (Δ der Relation)

$\text{verwandte}(\text{Vor}, \text{Nach}) :- \Delta\text{parent}(\text{Vor}, \text{Nach})$

$\text{verwandte}(\text{Vor}, \text{Nach}) :- \Delta\text{verwandte}(\text{Vor}, \text{Mitte}), \text{parent}(\text{Mitte}, \text{Nach})$

$\text{verwandte}(\text{Vor}, \text{Nach}) :- \text{verwandte}(\text{Vor}, \text{Mitte}), \Delta\text{parent}(\text{Mitte}, \text{Nach})$



$\text{verwandte}(\text{Vor}, \text{Nach}) :- \Delta\text{verwandte}(\text{Vor}, \text{Mitte}), \text{parent}(\text{Mitte}, \text{Nach})$



Deduktive Datenbanken

Semi-Naive Auswertung der Rekursion

parent \leq (K1,K11), (K1,K12), (K11,K111), (K11,K112), (K12,K121),
(K12,K122)

verwandte(Vor, Nach) :- Δ verwandte(Vor, Mitte), parent(Mitte, Nach)

Schritt 0: (K1,K11), (K1,K12), (K11,K111), (K11,K112), (K12,K121),
(K12,K122)

Schritt 1: (K1, K111), (K1, K112), (K1, K121), (K1, K122)

Schritt 2: \emptyset



Deduktive Datenbanken

Sichere Programme

- Durch Datalog-Regeln erzeugte Relationen müssen endlich sein
- Jede Variable, die in einer Regel vorkommt, muss eingeschränkt sein:
 - die Variable im Rumpf kommt in mindestens einem normalen Prädikat vor (nicht nur in Vergleichsprädikaten) oder
 - $X = c$ mit Konstante c existiert oder
 - ein Prädikat $X = Y$ vorkommt, und Y bereits eingeschränkt ist



Deduktive Datenbanken

Stratifizierte Programme

$$p(\dots) \text{ :- } q_1(\dots), \dots, \neg q_i(\dots), \dots, q_n(\dots).$$

- Regel p mit **negiertem** Prädikat q_i nur sinnvoll auswertbar, wenn q_i bereits materialisiert ist
- zuerst alle Regeln mit q_i auswerten $\Rightarrow q_i$ materialisiert
- nur möglich, wenn q_i nicht von p abhängig ist
- ➔ Abhängigkeitsgraph darf keine Pfade von p nach q_i enthalten
- muss für alle Regeln gelten



Aufgabe 2

Ist folgendes Datalog-Programm stratifiziert?

$$p(X, Y) \text{ :- } q_1(Y, Z), \neg q_2(Z, X), q_3(X, P).$$

$$q_2(Z, X) \text{ :- } q_4(Z, Y), q_3(Y, X).$$

$$q_4(Z, Y) \text{ :- } p(Z, X), q_3(X, Y).$$

Ist das Programm sicher – unter der Annahme, dass p, q_1, q_2, q_3, q_4 IDB- oder EDB-Prädikate sind?

Verteilte Datenbanksysteme

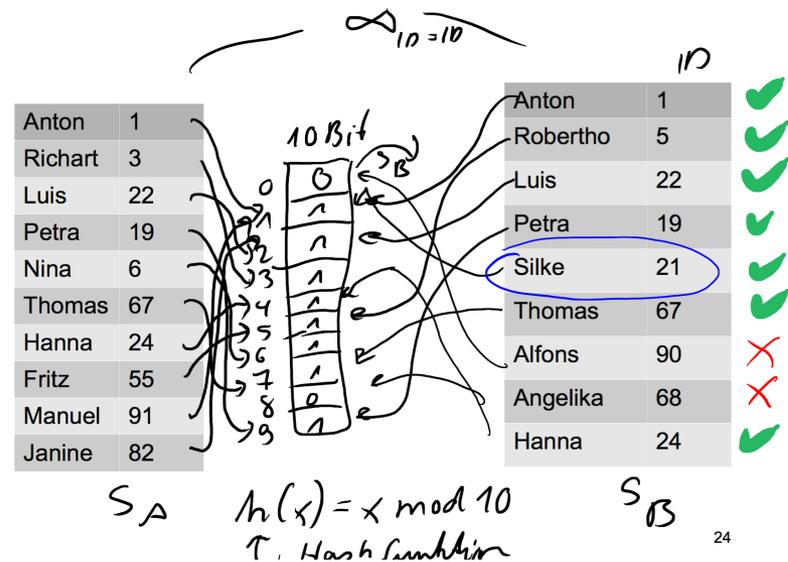
Bloom-Filter

- Einsatz bei sehr voluminösen Join-Attributen (z.B. lange Strings)

+ Verringerung der Transferkosten/
Netzwerkauslastung durch
Tupelvorauswahl mittels
Hashfunktion

+ Filter wird kompakter (Bitvektor V)

- Filterpräzision geht verloren





Verteilte Datenbanksysteme

Bloom-Filter

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V
0
1
2
3
4
5
6
7
8
9

S	
Raum	Gebäude
1	IMETUM
2	MI Büro
4	Physik
6	MW
7	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie

$$h(x) = x \bmod 10$$

Verteilte Datenbanksysteme

Bloom-Filter

1. Tabelle R mit $h(x)$ auf V mappen:

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5



V
0
1
2 1
3
4
5
6
7
8
9

S	
Raum	Gebäude
1	IMETUM
2	MI Büro
4	Physik
6	MW
7	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie

$$h(2) = 2 \bmod 10 = 2$$

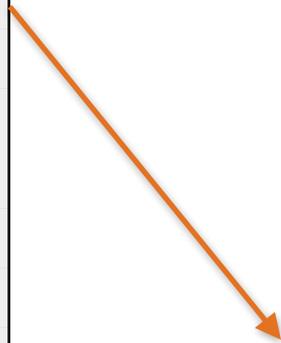
Verteilte Datenbanksysteme

Bloom-Filter

1. Tabelle R mit $h(x)$ auf V mappen:

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V
0
1
2 1
3
4
5
6
7 1
8
9



S	
Raum	Gebäude
1	IMETUM
2	MI Büro
4	Physik
6	MW
7	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie

$$h(7) = 7 \bmod 10 = 7$$

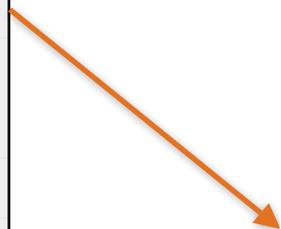
Verteilte Datenbanksysteme

Bloom-Filter

1. Tabelle R mit $h(x)$ auf V mappen:

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V
0
1
2 1
3
4
5
6 1
7 1
8
9



S	
Raum	Gebäude
1	IMETUM
2	MI Büro
4	Physik
6	MW
7	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie

$$h(6) = 6 \bmod 10 = 6$$

Verteilte Datenbanksysteme

Bloom-Filter

1. Tabelle R mit $h(x)$ auf V mappen:

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V
0
1
2 1
3
4
5
6 1
7 1
8
9



S	
Raum	Gebäude
1	IMETUM
2	MI Büro
4	Physik
6	MW
7	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie

$$h(2) = 2 \bmod 10 = 2$$

Verteilte Datenbanksysteme

Bloom-Filter

1. Tabelle R mit $h(x)$ auf V mappen:

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V
0
1
2 1
3
4
5
6 1
7 1
8 1
9



S	
Raum	Gebäude
1	IMETUM
2	MI Büro
4	Physik
6	MW
7	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie

$$h(8) = 8 \bmod 10 = 8$$

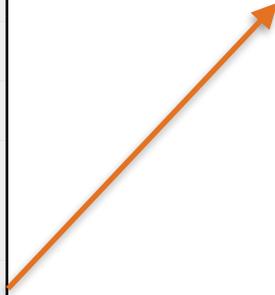
Verteilte Datenbanksysteme

Bloom-Filter

1. Tabelle R mit $h(x)$ auf V mappen:

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V
0
1 1
2 1
3
4
5
6 1
7 1
8 1
9



S	
Raum	Gebäude
1	IMETUM
2	MI Büro
4	Physik
6	MW
7	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie

$$h(1) = 1 \bmod 10 = 1$$

Verteilte Datenbanksysteme

Bloom-Filter

1. Tabelle R mit $h(x)$ auf V mappen:

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V
0
1 1
2 1
3
4
5
6 1
7 1
8 1
9



S	
Raum	Gebäude
1	IMETUM
2	MI Büro
4	Physik
6	MW
7	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie

$$h(16) = 16 \bmod 10 = 6$$

Verteilte Datenbanksysteme

Bloom-Filter

1. Tabelle R mit $h(x)$ auf V mappen:

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V
0
1 1
2 1
3
4
5
6 1
7 1
8 1
9

S	
Raum	Gebäude
1	IMETUM
2	MI Büro
4	Physik
6	MW
7	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie

$$h(8) = 8 \bmod 10 = 8$$

Verteilte Datenbanksysteme

Bloom-Filter

1. Tabelle R mit $h(x)$ auf V mappen:

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V
0
1 1
2 1
3
4
5
6 1
7 1
8 1
9

S	
Raum	Gebäude
1	IMETUM
2	MI Büro
4	Physik
6	MW
7	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie

$$h(7) = 7 \bmod 10 = 7$$

Verteilte Datenbanksysteme

Bloom-Filter

1. Tabelle R mit $h(x)$ auf V mappen:

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V
0
1 1
2 1
3
4
5 1
6 1
7 1
8 1
9

S	
Raum	Gebäude
1	IMETUM
2	MI Büro
4	Physik
6	MW
7	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie

$$h(5) = 5 \bmod 10 = 5$$



Verteilte Datenbanksysteme

Bloom-Filter

2. Felder in V ohne hash-Treffer mit 0 füllen

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V	
0	0
1	1
2	1
3	0
4	0
5	1
6	1
7	1
8	1
9	0

S	
Raum	Gebäude
1	IMETUM
2	MI Büro
4	Physik
6	MW
7	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie

$$h(x) = x \bmod 10$$



Verteilte Datenbanksysteme

Bloom-Filter

3. Bitvektor V an S schicken

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V	
0	0
1	1
2	1
3	0
4	0
5	1
6	1
7	1
8	1
9	0

S	
Raum	Gebäude
1	IMETUM
2	MI Büro
4	Physik
6	MW
7	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie

$$h(x) = x \bmod 10$$



Verteilte Datenbanksysteme

Bloom-Filter

4. S überprüft mit $h(x)$ den Bitvektor V

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V
0 0
1 1
2 1
3 0
4 0
5 1
6 1
7 1
8 1
9 0

S	
Raum	Gebäude
1	IMETUM
2	MI Büro
4	Physik
6	MW
7	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie

$$h(x) = x \bmod 10$$

Verteilte Datenbanksysteme

Bloom-Filter

4. S überprüft mit $h(x)$ den Bitvektor V

- ✓ Tupel wird zur Station mit R geschickt
- ✗ Tupel wird nicht übermittelt

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V	
0	0
1	1
2	1
3	0
4	0
5	1
6	1
7	1
8	1
9	0



S	
Raum	Gebäude
1	IMETUM
2	MI Büro
4	Physik
6	MW
7	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie

$$h(1) = 1 \bmod 10 = 1$$

Verteilte Datenbanksysteme

Bloom-Filter

4. S überprüft mit $h(x)$ den Bitvektor V

- ✓ Tupel wird zur Station mit R geschickt
- ✗ Tupel wird nicht übermittelt

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V	
0	0
1	1
2	1
3	0
4	0
5	1
6	1
7	1
8	1
9	0



S	
Raum	Gebäude
1	IMETUM
2	MI Büro
4	Physik
6	MW
7	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie

$$h(2) = 2 \bmod 10 = 2$$

Verteilte Datenbanksysteme

Bloom-Filter

4. S überprüft mit $h(x)$ den Bitvektor V

- ✓ Tupel wird zur Station mit R geschickt
- ✗ Tupel wird nicht übermittelt

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V	
0	0
1	1
2	1
3	0
4	0
5	1
6	1
7	1
8	1
9	0

S	
Raum	Gebäude
1	IMETUM
2	MI Büro
4	Physik
6	MW
7	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie



$$h(4) = 4 \bmod 10 = 4$$



Verteilte Datenbanksysteme

Bloom-Filter

4. S überprüft mit $h(x)$ den Bitvektor V

- ✓ Tupel wird zur Station mit R geschickt
- ✗ Tupel wird nicht übermittelt

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V
0 0
1 1
2 1
3 0
4 0
5 1
6 1
7 1
8 1
9 0

S	
Raum	Gebäude
1 ✓	IMETUM
2 ✓	MI Büro
4 ✗	Physik
6 ✓	MW
7	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie

$$h(6) = 6 \bmod 10 = 6$$



Verteilte Datenbanksysteme

Bloom-Filter

4. S überprüft mit $h(x)$ den Bitvektor V

- ✓ Tupel wird zur Station mit R geschickt
- ✗ Tupel wird nicht übermittelt

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V
0 0
1 1
2 1
3 0
4 0
5 1
6 1
7 1
8 1
9 0

S	
Raum	Gebäude
1 ✓	IMETUM
2 ✓	MI Büro
4 ✗	Physik
6 ✓	MW
7 ✓	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie

$$h(7) = 7 \bmod 10 = 7$$

Verteilte Datenbanksysteme

Bloom-Filter

4. S überprüft mit $h(x)$ den Bitvektor V

- ✓ Tupel wird zur Station mit R geschickt
- ✗ Tupel wird nicht übermittelt

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V	
0	0
1	1
2	1
3	0
4	0
5	1
6	1
7	1
8	1
9	0

S		
Raum		Gebäude
1	✓	IMETUM
2	✓	MI Büro
4	✗	Physik
6	✓	MW
7	✓	MI Raum
8	✓	ERI
9		MI Bib
10		Physik
11		Chemie

$$h(8) = 8 \bmod 10 = 8$$

Verteilte Datenbanksysteme

Bloom-Filter

4. S überprüft mit $h(x)$ den Bitvektor V

- ✓ Tupel wird zur Station mit R geschickt
- ✗ Tupel wird nicht übermittelt

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V	
0	0
1	1
2	1
3	0
4	0
5	1
6	1
7	1
8	1
9	0

S		
Raum		Gebäude
1	✓	IMETUM
2	✓	MI Büro
4	✗	Physik
6	✓	MW
7	✓	MI Raum
8	✓	ERI
9	✗	MI Bib
10		Physik
11		Chemie

$h(9) = 9 \bmod 10 = 9$

Verteilte Datenbanksysteme

Bloom-Filter

4. S überprüft mit $h(x)$ den Bitvektor V

- ✓ Tupel wird zur Station mit R geschickt
- ✗ Tupel wird nicht übermittelt

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V	
0	0
1	1
2	1
3	0
4	0
5	1
6	1
7	1
8	1
9	0

S		
Raum		Gebäude
1	✓	IMETUM
2	✓	MI Büro
4	✗	Physik
6	✓	MW
7	✓	MI Raum
8	✓	ERI
9	✗	MI Bib
10	✗	Physik
11		Chemie

$$h(10) = 10 \bmod 10 = 0$$



Verteilte Datenbanksysteme

Bloom-Filter

False positive

4. S überprüft mit $h(x)$ den Bitvektor V

- ✓ Tupel wird zur Station mit R geschickt
- ✗ Tupel wird nicht übermittelt

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V
0 0
1 1
2 1
3 0
4 0
5 1
6 1
7 1
8 1
9 0

S		
Raum		Gebäude
1	✓	IMETUM
2	✓	MI Büro
4	✗	Physik
6	✓	MW
7	✓	MI Raum
8	✓	ERI
9	✗	MI Bib
10	✗	Physik
11	✓	Chemie

$$h(11) = 11 \bmod 10 = 1$$



Verteilte Datenbanksysteme

Bloom-Filter

5. Übermitteln der Treffer zur Station R

- ✓ Tupel wird zur Station mit R geschickt
- ✗ Tupel wird nicht übermittelt

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V	
0	0
1	1
2	1
3	0
4	0
5	1
6	1
7	1
8	1
9	0

S	
Raum	Gebäude
1	✓ IMETUM
2	✓ MI Büro
4	✗ Physik
6	✓ MW
7	✓ MI Raum
8	✓ ERI
9	✗ MI Bib
10	✗ Physik
11	✓ Chemie

$$h(x) = x \bmod 10$$



Verteilte Datenbanksysteme

Bloom-Filter

False positives werden übermittelt und von R beim Join verworfen.

False positive Rate
1/6

- ✓ Tupel wird zur Station mit R geschickt
- ✗ Tupel wird nicht übermittelt

R	
Pers	Raum
Max	2
Magda	7
Tom	6
Alex	2
Julius	8
Kathi	1
Anna	16
Gregor	8
Thuy	7
Domi	5

V	
0	0
1	1
2	1
3	0
4	0
5	1
6	1
7	1
8	1
9	0

S		
Raum		Gebäude
1	✓	IMETUM
2	✓	MI Büro
4	✗	Physik
6	✓	MW
7	✓	MI Raum
8	✓	ERI
9	✗	MI Bib
10	✗	Physik
11	✓	Chemie

$$h(x) = x \bmod 10$$



Aufgabe 3

Gegeben seien die Tabellen **Studenten** und **Punkte** mit Schlüssel **MatrNr**, wobei **Punkte** auf einem separaten Rechner gespeichert ist. Es soll folgende Anfrage ausgeführt werden:

```
SELECT Name, Bonus FROM Student s, Punkte p WHERE s.MatrNr = p.MatrNr;
```

Der Datenbankadministrator entscheidet sich für einen Bloom-Filter zur Vorauswahl der Tupel. Auf **MatrNr** wird die Hash-Funktion $h(x) = x \bmod 5$ angewendet.

Studenten			Punkte		
<u>MatrNr</u>	Name	Hashwert	<u>MatrNr</u>	Bonus	Hashwert
27	Magda		27	ja	
4	Josef		16	nein	
19	Erik		25	nein	
95	Philipp		95	ja	

- Berechnen Sie die Hash-Werte und tragen Sie diese in die obige Tabelle ein.
- Füllen Sie den von **Studenten** zu übertragenden Bitvektor aus. Verwenden Sie 0 oder 1.
- Geben Sie basierend auf dem Bitvektor an, welche Tupel aus **Punkte** übertragen werden (nur **MatrNr** angeben).
- Geben Sie die Falsch-Positiv-Rate (false positive rate) an.
- Nehmen Sie an, dass jedes Tupel 8 Byte und der Bloomfilter selbst 1 Byte groß ist. Berechnen Sie zunächst die übertragenen Bytes ohne und mit Einsatz des Bloom-Filters.



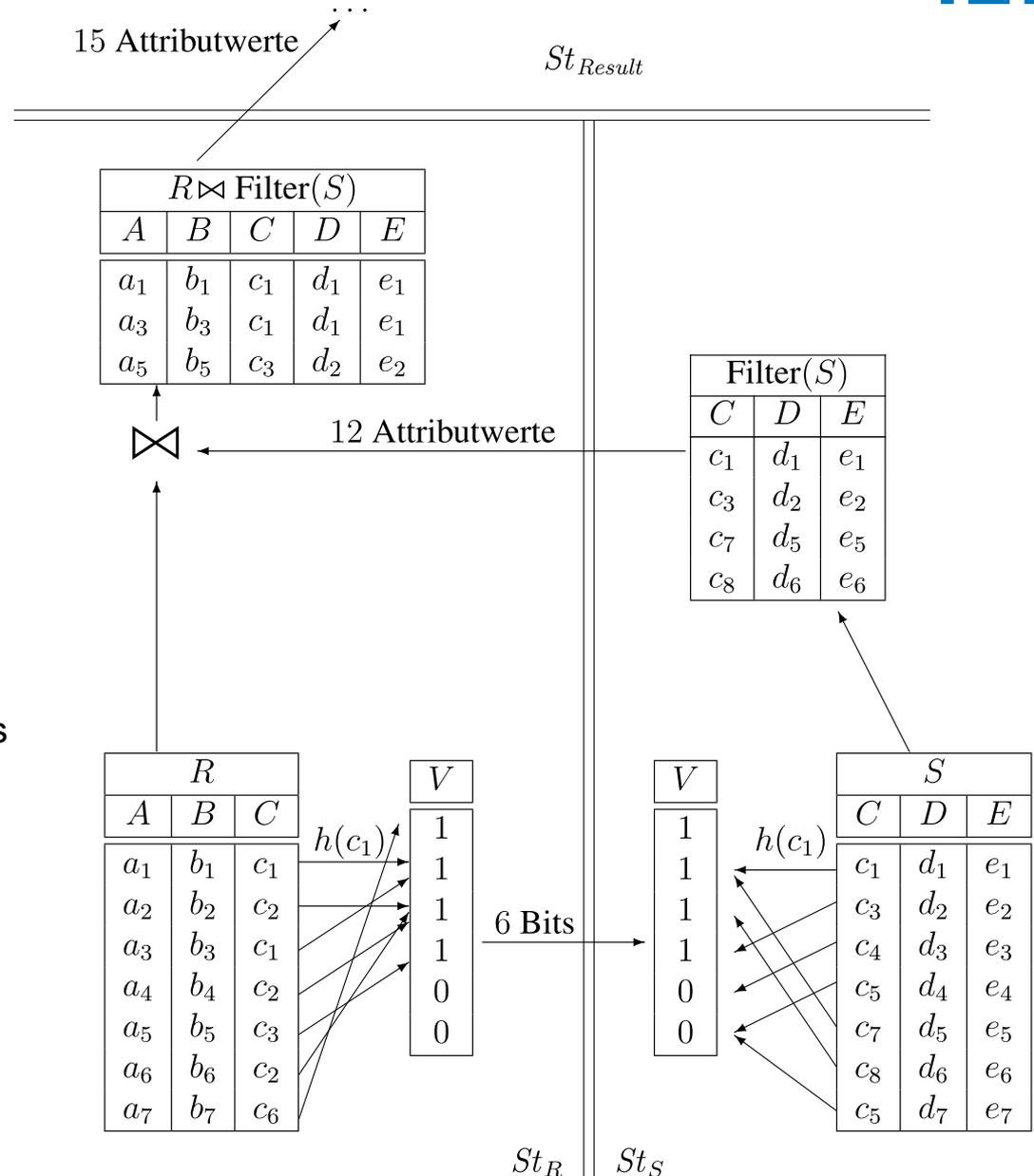
Aufgabe 4

Überlegen Sie sich, welche Tupel bei der Anwendung des bloomfilterbasierten Joins in Abbildung 2 übertragen werden.

Markieren Sie insbesondere, welche Tupel übertragen werden, obwohl sie keinen Joinpartner finden (sog. false drops).

Wie kann die Anzahl dieser false drops verringert werden?

Welche Eigenschaften sollte die Hashfunktion $h(c)$, die bei dieser Joinbearbeitung verwendet wird, erfüllen?





Fragen?