



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.

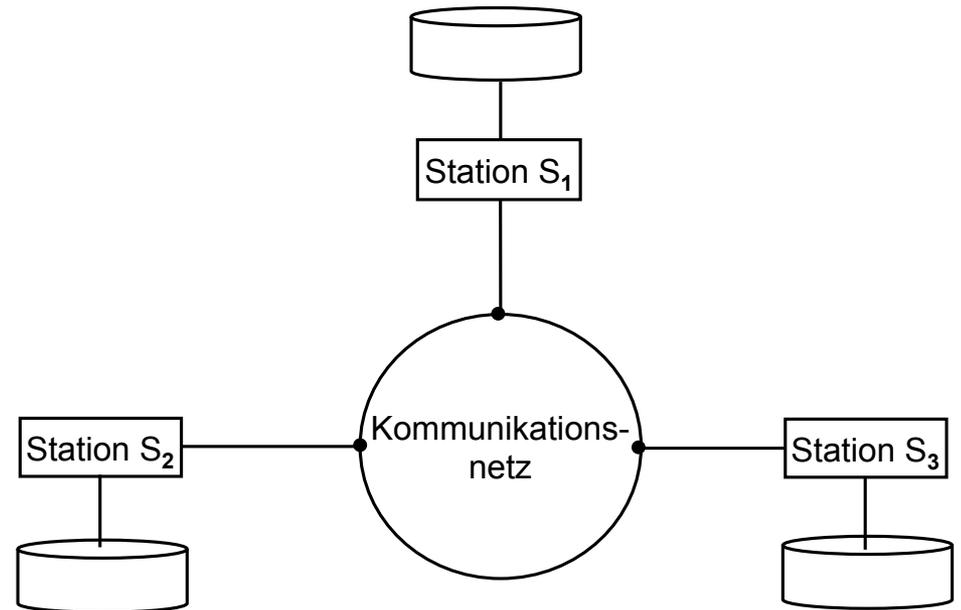


Verteilte Datenbanken

Verteilte Datenbanksysteme

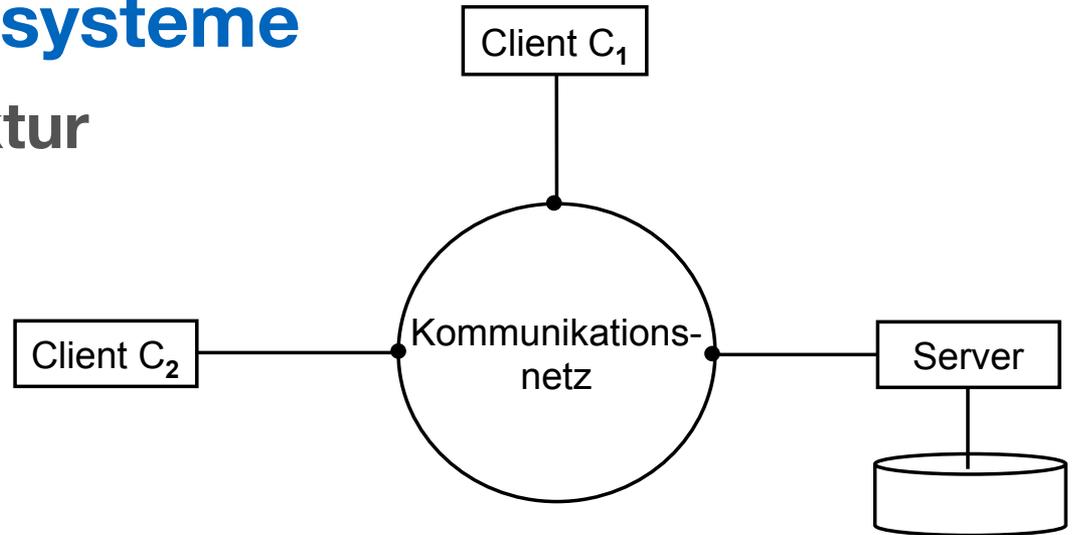
Motivation

- Globale Gesamtinformation auf Stationen (Sites) verteilt
- Daten werden von verteilten Datenbankverwaltungssystemen (VDBMS) verwaltet
- Stationen dürfen lokale Daten bearbeiten
- Kommunikationsverbindung (LAN, WAN, Telefonverbindungen)

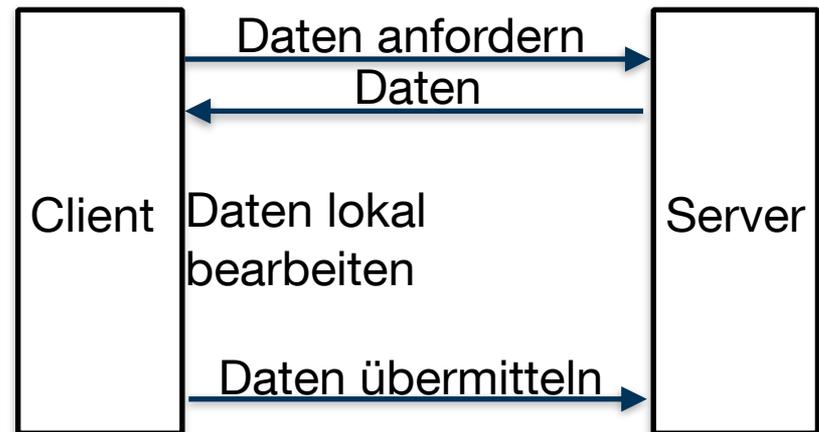


Verteilte Datenbanksysteme

Client-/Server-Architektur



- Degradiertes verteiltes Datenbanksystem
- Nur Server darf Daten abspeichern





Verteilte Datenbanksysteme

Fragmentierung

Horizontale Fragmentierung:

- Relation wird in disjunkte Tupelmengen geteilt

Vertikale Fragmentierung:

- Relation wird nach Attributen geteilt (durch Projektionen)
- Kombinierte Fragmentierung möglich



Aufgabe 1

Professoren						
PersNr	Name	Rang	Raum	Fakultät	Gehalt	Steuerklasse
2125	Sokrates	C4	226	Philosophie	85000	1
2126	Russel	C4	232	Philosophie	80000	3
2127	Kopernikus	C3	310	Physik	65000	5
2133	Popper	C3	52	Philosophie	68000	1
2134	Augustinus	C3	309	Theologie	55000	5
2136	Curie	C4	36	Physik	95000	3
2137	Kant	C4	7	Philosophie	98000	1

Gehen Sie von folgender kombinierter Fragmentierung der in Abbildung 1 dargestellten Relation *Professoren* aus:

1. Zuerst erfolgt eine vertikale Fragmentierung in

$$\text{ProfVerw} := \Pi_{\text{PersNr, Name, Gehalt, Steuerklasse}}(\text{Professoren})$$
$$\text{Profs} := \Pi_{\text{PersNr, Name, Rang, Raum, Fakultät}}(\text{Professoren})$$

2. Das Fragment Profs wird weiter horizontal fragmentiert in

$$\text{TheolProfs} := \sigma_{\text{Fakultät} = \text{'Theologie'}}(\text{Profs})$$
$$\text{PhysikProfs} := \sigma_{\text{Fakultät} = \text{'Physik'}}(\text{Profs})$$
$$\text{PhiloProfs} := \sigma_{\text{Fakultät} = \text{'Philosophie'}}(\text{Profs})$$

Übersetzen Sie aufbauend auf dieser Fragmentierung die folgende SQL-Anfrage in die kanonische Form.

```
select Name, Gehalt, Rang
from Professoren
where Gehalt > 80000;
```

Optimieren Sie diesen kanonischen Auswertungsplan durch Anwendung algebraischer Transformationsregeln (Äquivalenzen).



Verteilte Datenbanksysteme

Fragmentierung

Professoren						
PersNr	Name	Rang	Raum	Fakultät	Gehalt	Steuerklass
2125	Sokrates	C4	226	Philosophie	85000	1
2126	Russel	C4	232	Philosophie	80000	3
2127	Kopernikus	C3	310	Physik	65000	5
2133	Popper	C3	52	Philosophie	68000	1
2134	Augustinus	C3	309	Theologie	55000	5
2136	Curie	C4	36	Physik	95000	3
2137	Kant	C4	7	Philosophie	98000	1

1. Zuerst erfolgt eine vertikale Fragmentierung in

$$\text{ProfVerw} := \Pi_{\text{PersNr, Name, Gehalt, Steuerklasse}}(\text{Professoren})$$
$$\text{Profs} := \Pi_{\text{PersNr, Name, Rang, Raum, Fakultät}}(\text{Professoren})$$



Verteilte Datenbanksysteme

Fragmentierung Vertikal



ProfVerw				Profs				
PersNr	Name	Gehalt	Steuerklasse	PersNr	Name	Rang	Raum	Fakultät
2125	Sokrates	85000	1	2125	Sokrates	C4	226	Philosophie
2126	Russel	80000	3	2126	Russel	C4	232	Philosophie
2127	Kopernikus	65000	5	2127	Kopernikus	C3	310	Physik
2133	Popper	68000	1	2133	Popper	C3	52	Philosophie
2134	Augustinus	55000	5	2134	Augustinus	C3	309	Theologie
2136	Curie	95000	3	2136	Curie	C4	36	Physik
2137	Kant	98000	1	2137	Kant	C4	7	Philosophie

2. Das Fragment Profs wird weiter horizontal fragmentiert in

TheolProfs := $\sigma_{\text{Fakultät} = \text{'Theologie'}}(\text{Profs})$

PhysikProfs := $\sigma_{\text{Fakultät} = \text{'Physik'}}(\text{Profs})$

PhiloProfs := $\sigma_{\text{Fakultät} = \text{'Philosophie'}}(\text{Profs})$



Verteilte Datenbanksysteme

Fragmentierung Horizontal

Profs				
PersNr	Name	Rang	Raum	Fakultät
2125	Sokrates	C4	226	Philosophie
2126	Russel	C4	232	Philosophie
2127	Kopernikus	C3	310	Physik
2133	Popper	C3	52	Philosophie
2134	Augustinus	C3	309	Theologie
2136	Curie	C4	36	Physik
2137	Kant	C4	7	Philosophie

2. Das Fragment Profs wird weiter horizontal fragmentiert in

TheolProfs := $\sigma_{\text{Fakultät} = \text{'Theologie'}}(\text{Profs})$

PhysikProfs := $\sigma_{\text{Fakultät} = \text{'Physik'}}(\text{Profs})$

PhiloProfs := $\sigma_{\text{Fakultät} = \text{'Philosophie'}}(\text{Profs})$



Verteilte Datenbanksysteme

Fragmentierung Horizontal

TheolProfs				
PersNr	Name	Rang	Raum	Fakultät
2134	Augustinus	C3	309	Theologie

PhysikProfs				
PersNr	Name	Rang	Raum	Fakultät
2127	Kopernikus	C3	310	Physik
2136	Curie	C4	36	Physik

PhiloProfs				
PersNr	Name	Rang	Raum	Fakultät
2125	Sokrates	C4	226	Philosophie
2126	Russel	C4	232	Philosophie
2133	Popper	C3	52	Philosophie
2137	Kant	C4	7	Philosophie

Zerlegung
in disjunkte
Tupelmengen

Verteilte Datenbanksysteme

Fragmentierung Vertikal & Horizontal

ProfVerw			
PersNr	Name	Gehalt	Steuerklasse
2125	Sokrates	85000	1
2126	Russel	80000	3
2127	Kopernikus	65000	5
2133	Popper	68000	1
2134	Augustinus	55000	5
2136	Curie	95000	3
2137	Kant	98000	1

TheolProfs				
PersNr	Name	Rang	Raum	Fakultät
2134	Augustinus	C3	309	Theologie

PhysikProfs				
PersNr	Name	Rang	Raum	Fakultät
2127	Kopernikus	C3	310	Physik
2136	Curie	C4	36	Physik

PhiloProfs				
PersNr	Name	Rang	Raum	Fakultät
2125	Sokrates	C4	226	Philosophie
2126	Russel	C4	232	Philosophie
2133	Popper	C3	52	Philosophie
2137	Kant	C4	7	Philosophie

Übersetzen Sie aufbauend auf dieser Fragmentierung die folgende SQL-Anfrage in die kanonische Form.

```
select Name, Gehalt Rang
from Professoren
where Gehalt > 80000;
```



Aufgabe 2

Für die Rekonstruierbarkeit der Originalrelation R aus vertikalen Fragmenten R_1, \dots, R_n reicht es eigentlich, wenn Fragmente paarweise einen Schlüsselkandidaten enthalten. Illustrieren Sie, warum es also nicht notwendig ist, dass der Durchschnitt aller Fragmentschemata einen Schlüsselkandidaten enthält. Es muss also nicht unbedingt gelten

$$R_1 \cap \dots \cap R_n \supseteq \kappa,$$

wobei κ ein Schlüsselkandidat aus R ist.

Geben Sie ein anschauliches Beispiel hierfür – am besten bezogen auf unsere Beispiel-Relation *Professoren*.



Aufgabe 3

<u>MatrNr</u>	Name	Note	Standort
10101	Philipp	1,0	München
10102	Magdalena	1,0	Garching
10103	Erik	1,0	Garching
10104	Josef	1,0	Garching
10105	Alex	1,0	Garching
10106	Maxmilian	1,0	München

Für eine verteilte Datenbank soll die Tabelle geeignet fragmentiert werden. Ziel ist, Namen mit Standort der Studenten lokal und die Noten getrennt abzupeichern.

- 1) Fragmentieren Sie die Relation geeignet *vertikal*.
 - a) Geben Sie das Schema für die zwei resultierenden Relationen $KlausurV_1$ und $KlausurV_2$ an. Unterstreichen Sie jeweils den Primärschlüssel.
 - b) Geben Sie in SQL-92 die zwei resultierenden Relationen $KlausurV_1$ und $KlausurV_2$ als Hilfstabellen (mittels `with`) an.
- 2) Die geeignetere der beiden resultierenden Relationen soll *horizontal* fragmentiert werden.
 - a) Geben Sie das Prädikat der Selektion an, mit dem fragmentiert wird.
 - b) Geben Sie in SQL-92 die zwei resultierenden Relationen $KlausurH_1$ und $KlausurH_2$ als Hilfstabellen (mittels `with`) an.
- 3) Schreiben Sie eine SQL-Abfrage, die die Ursprungsrelation aus den Teilrelationen zusammensetzt.



Verteilte Datenbanksysteme

Quorum-Consensus Verfahren

- Ausgleich der Leistungsfähigkeit zwischen Lese- und Änderungstransaktionen
- ➔ Teilweise Verlagerung des Overheads von Änderungs- zu Lesetransaktionen:
 - Kopien A_i von A werden individuelle Gewichte zugeordnet
- Lesequorum $Q_r(A)$
- Schreibquorum $Q_w(A)$

Folgende Bedingungen müssen gelten:

1. $Q_w(A) + Q_w(A) > W(A)$
2. $Q_r(A) + Q_w(A) > W(A)$

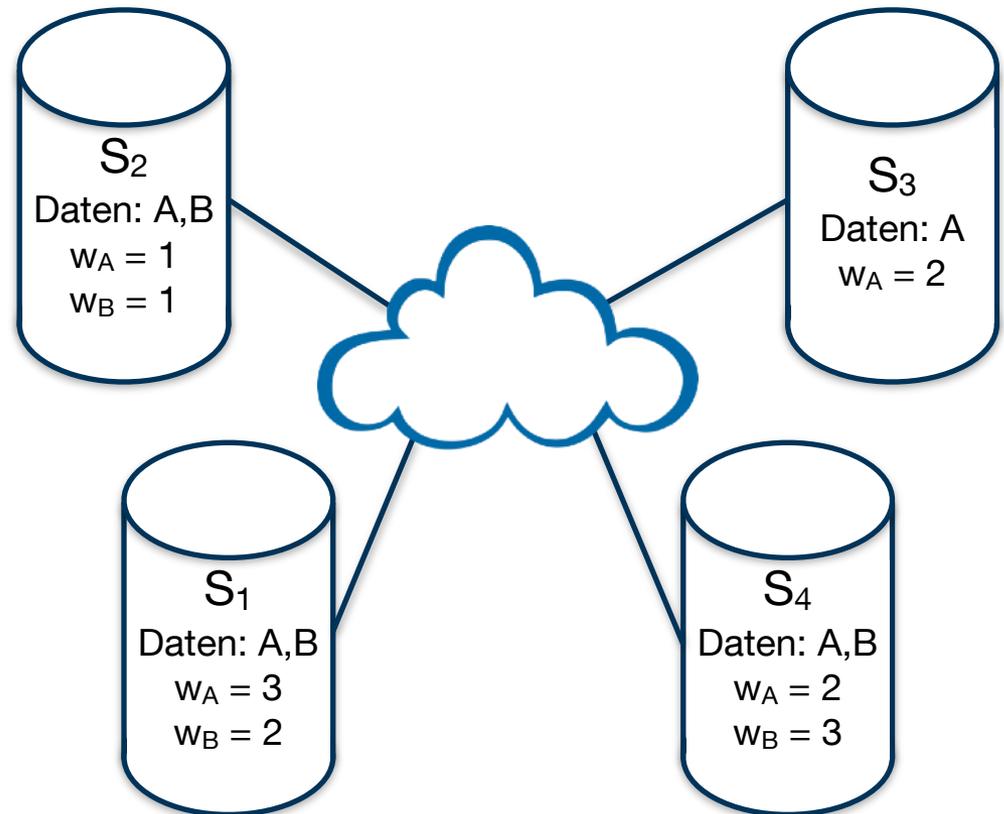
Verteilte Datenbanksysteme

Quorum-Consensus Verfahren

Berechne das Schreib- Q_w und Lesequorum Q_r für A und B.

$$W(A) = \sum_{i=1} w_i(A)$$

1. $Q_w(A) + Q_w(A) > W(A)$
2. $Q_r(A) + Q_w(A) > W(A)$



Verteilte Datenbanksysteme

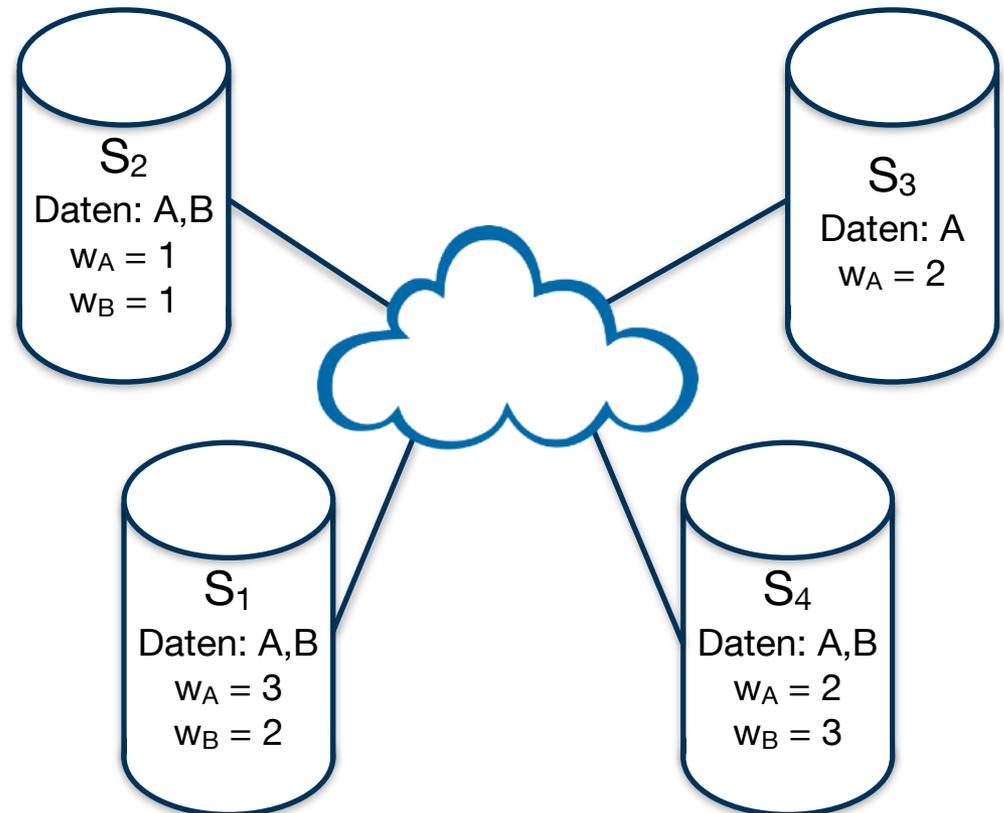
Quorum-Consensus Verfahren

Berechne das Schreib- Q_w und Lesequorum Q_r für A und B.

Für A:

1. Gesamtgewicht berechnen:

$$W(A) = 3+1+2+2 = 8$$



Verteilte Datenbanksysteme

Quorum-Consensus Verfahren

Berechne das Schreib- Q_w und Lesequorum Q_r für A und B.

Für A:

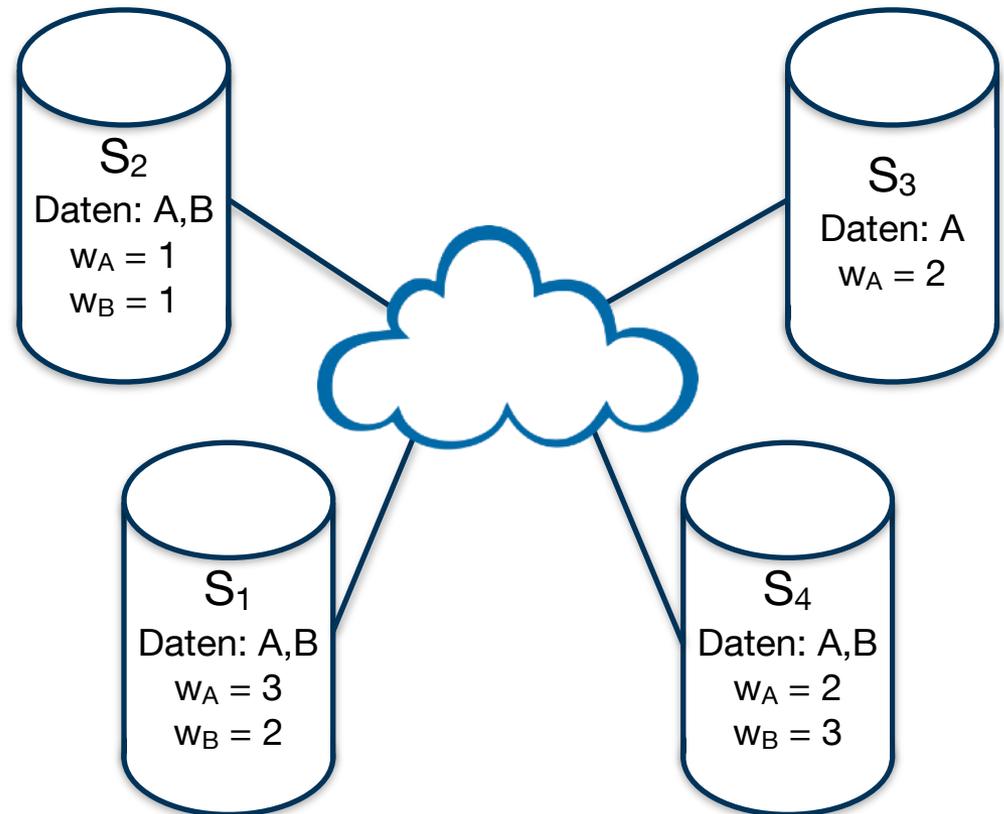
2. Schreibquorum berechnen

$$Q_w(A) + Q_w(A) > W(A)$$

$$\Rightarrow 2 * Q_w(A) > 8 \quad |:2$$

$$\Rightarrow Q_w(A) > 4$$

$$\Rightarrow Q_w(A) = 5$$



Verteilte Datenbanksysteme

Quorum-Consensus Verfahren

Berechne das Schreib- Q_w und Lesequorum Q_r für A und B.

Für A:

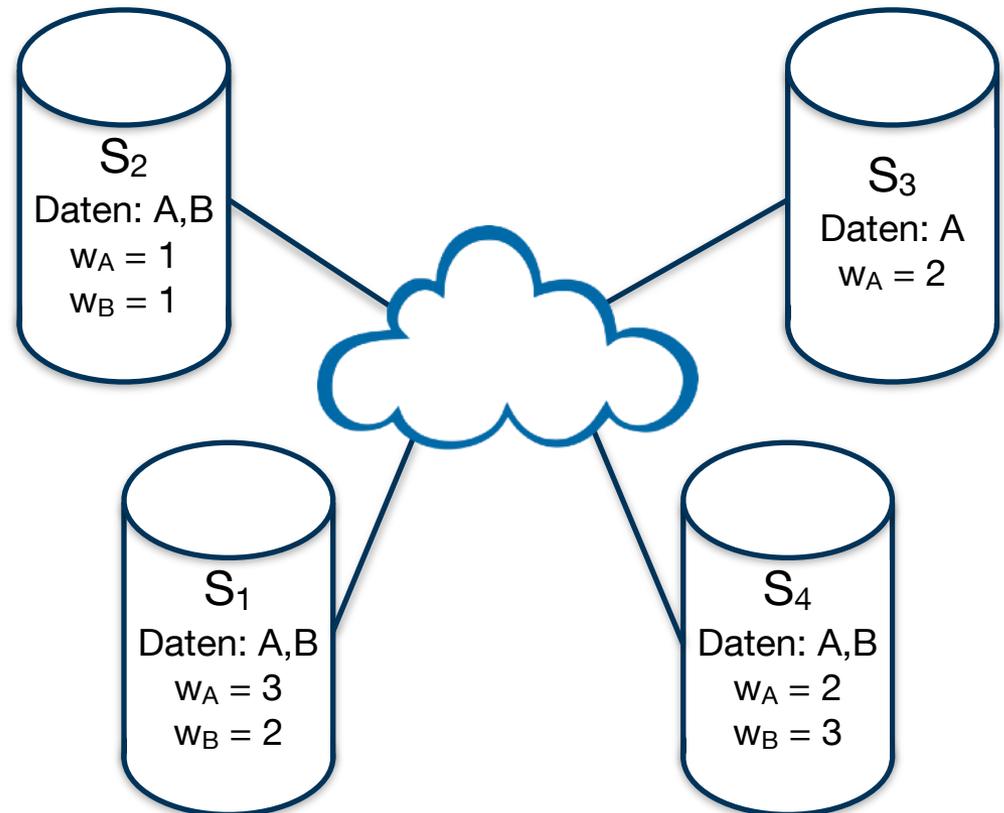
2. Lesequorum berechnen

$$Q_r(A) + Q_w(A) > W(A)$$

$$\Rightarrow Q_r(A) + 5 > 8 \quad | -5$$

$$\Rightarrow Q_r(A) > 3$$

$$\Rightarrow Q_r(A) = 4$$



Verteilte Datenbanksysteme

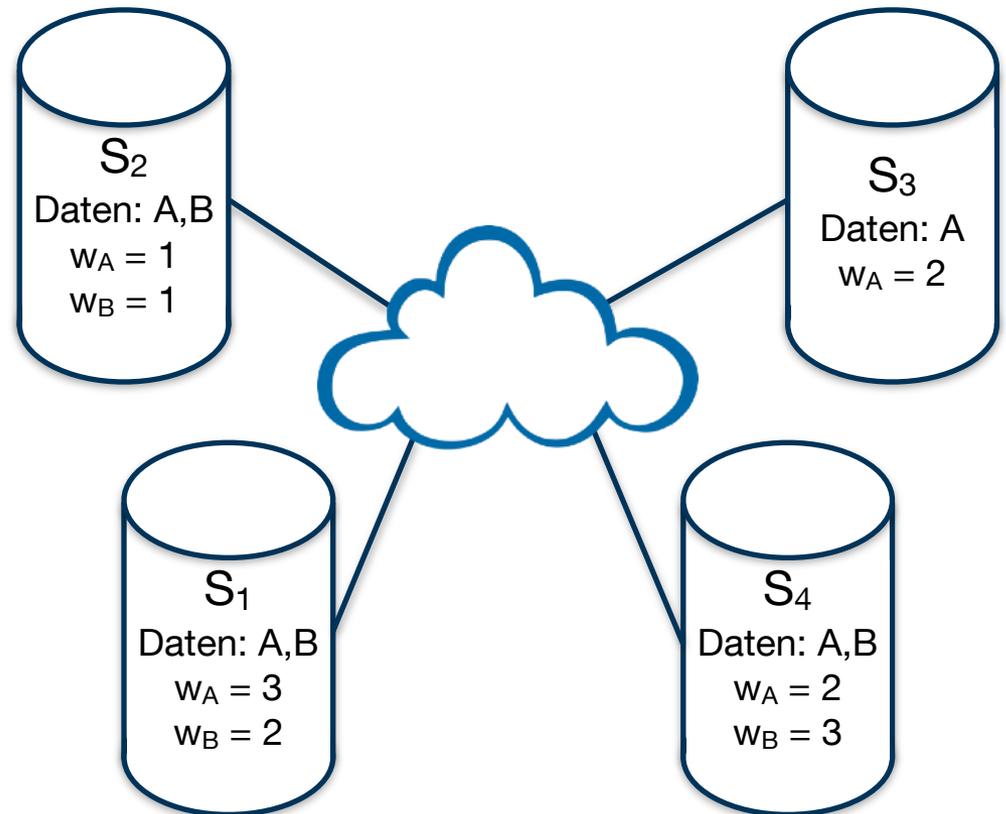
Quorum-Consensus Verfahren

Berechne das Schreib- Q_w und Lesequorum Q_r für A und B.

Für B:

1. Gesamtgewicht berechnen:

$$W(B) = 2+1+3 = 6$$



Verteilte Datenbanksysteme

Quorum-Consensus Verfahren

Berechne das Schreib- Q_w und Lesequorum Q_r für A und B.

Für B:

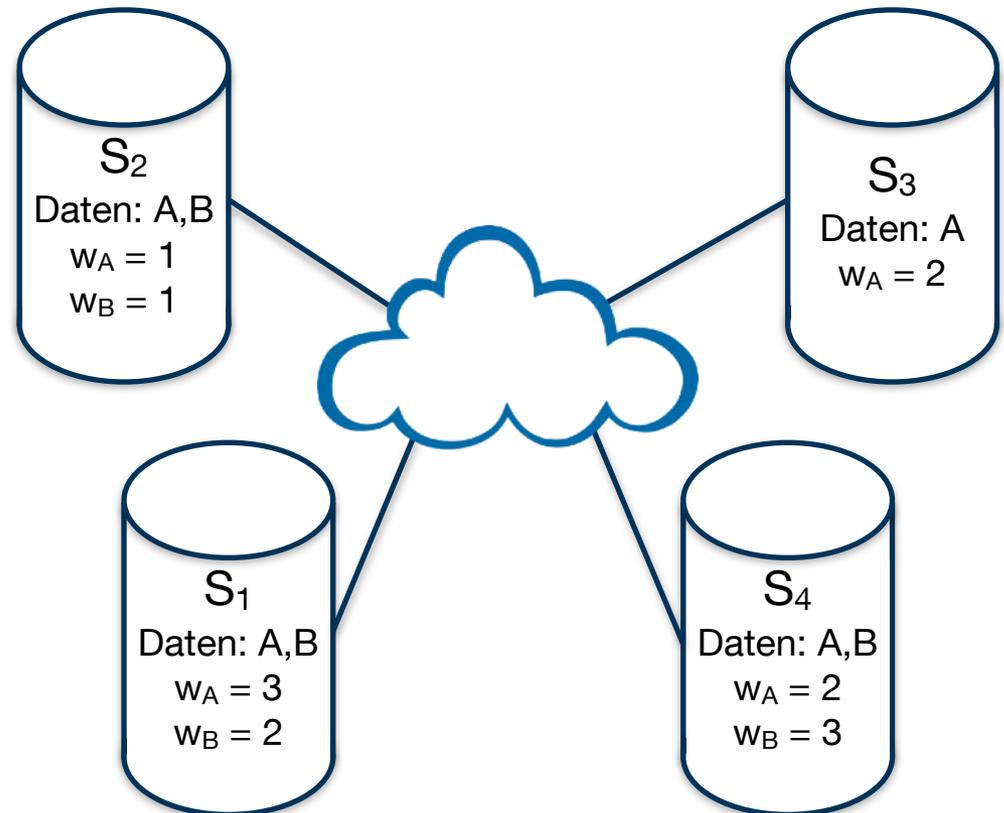
2. Schreibquorum berechnen

$$Q_w(B) + Q_w(B) > W(B)$$

$$\Rightarrow 2 * Q_w(B) > 6 \quad |:2$$

$$\Rightarrow Q_w(B) > 3$$

$$\Rightarrow Q_w(B) = 4$$



Verteilte Datenbanksysteme

Quorum-Consensus Verfahren

Berechne das Schreib- Q_w und Lesequorum Q_r für A und B.

Für B:

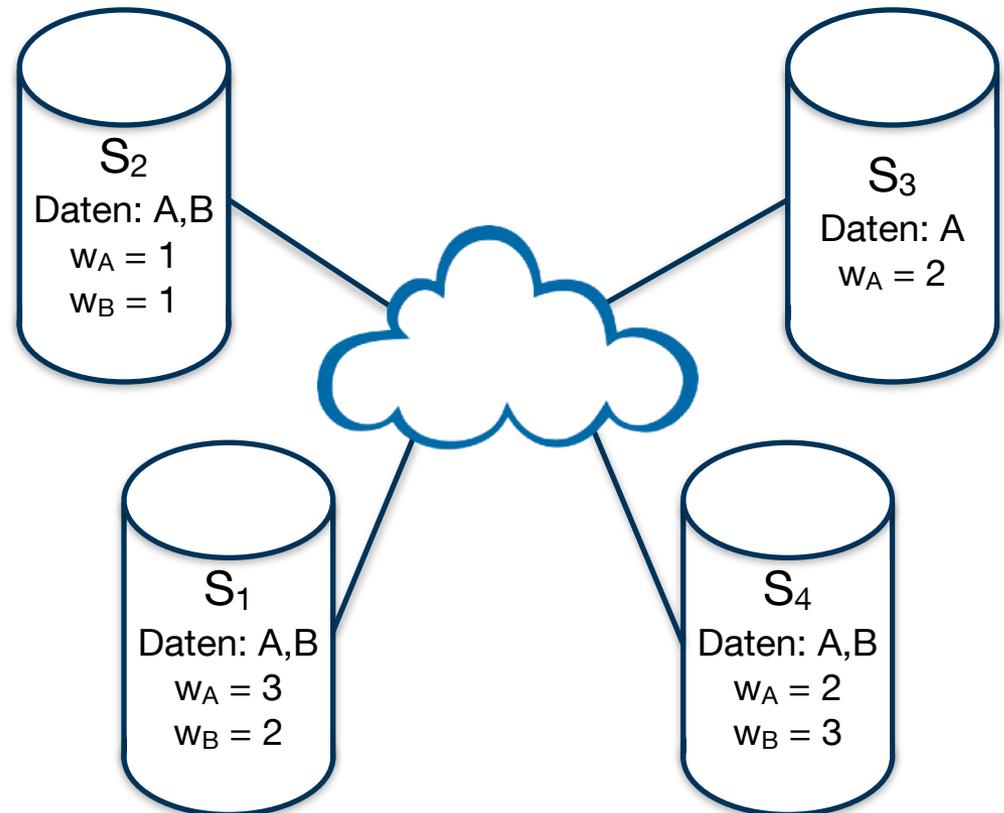
2. Lesequorum berechnen

$$Q_r(B) + Q_w(B) > W(B)$$

$$\Rightarrow Q_r(B) + 4 > 6 \quad |-4$$

$$\Rightarrow Q_r(B) > 2$$

$$\Rightarrow Q_r(B) = 3$$



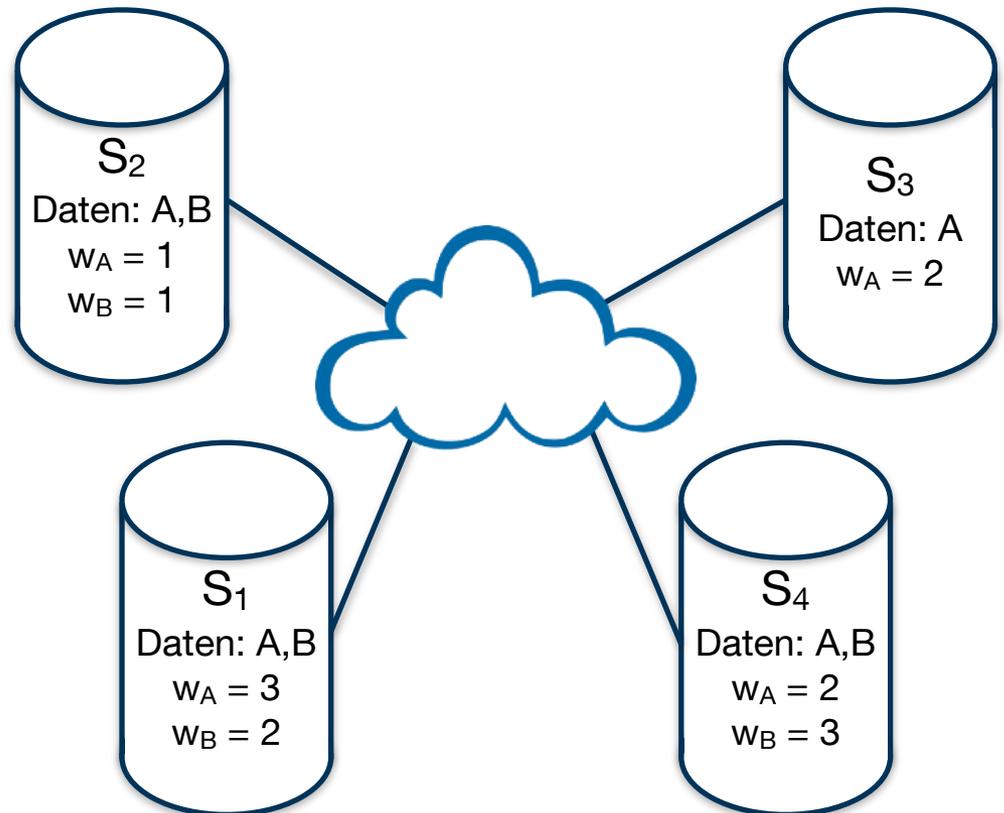
Verteilte Datenbanksysteme

Quorum-Consensus Verfahren

Schreibe:

- $A^- = 5$
- $B^+ = 2$

Station	Daten	Gewichte	Version
S ₁	A = 22 B = 7	w _A = 3 w _B = 2	A ₁ B ₁
S ₂	A = 22 B = 7	w _A = 1 w _B = 1	A ₁ B ₁
S ₃	A = 22	w _A = 2	A ₁
S ₄	A = 22 B = 7	w _A = 2 w _B = 3	A ₁ B ₁



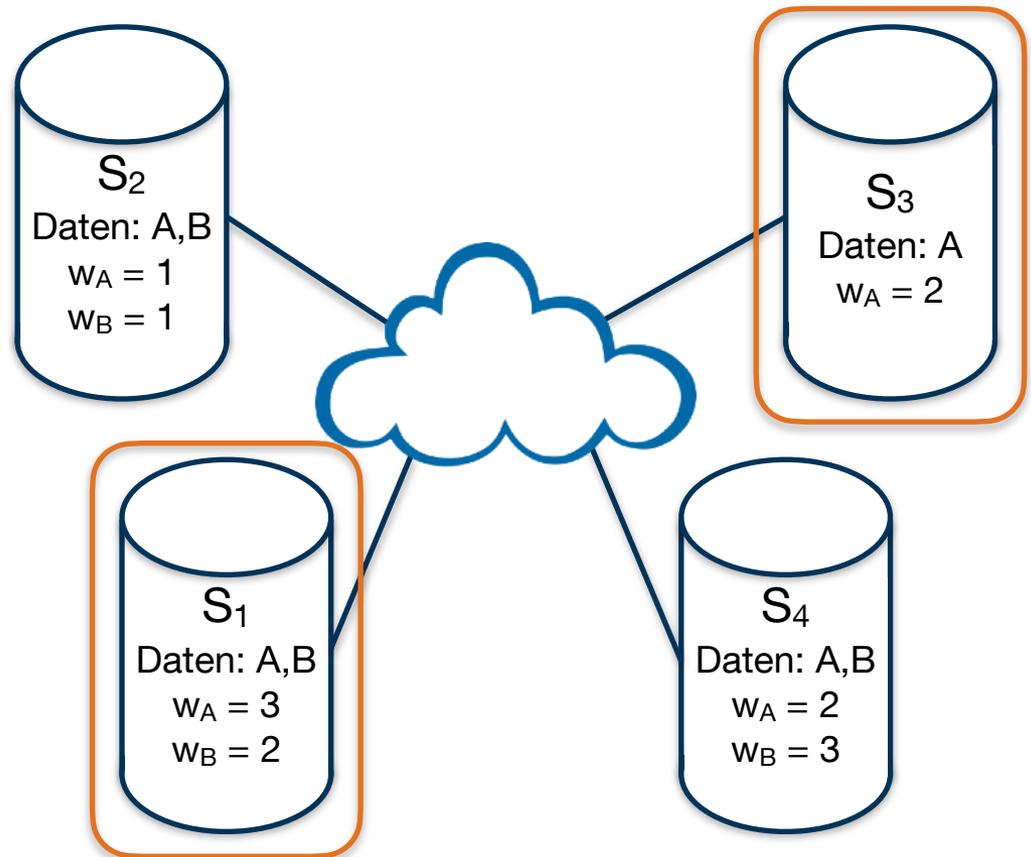
Verteilte Datenbanksysteme

Quorum-Consensus Verfahren

Schreibe:

- $A_- = 5$
- $B_+ = 2$

Station	Daten	Gewichte	Version
S ₁	A = 22 B = 7	w _A = 3 w _B = 2	A ₁ B ₁
S ₂	A = 22 B = 7	w _A = 1 w _B = 1	A ₁ B ₁
S ₃	A = 22	w _A = 2	A ₁
S ₄	A = 22 B = 7	w _A = 2 w _B = 3	A ₁ B ₁



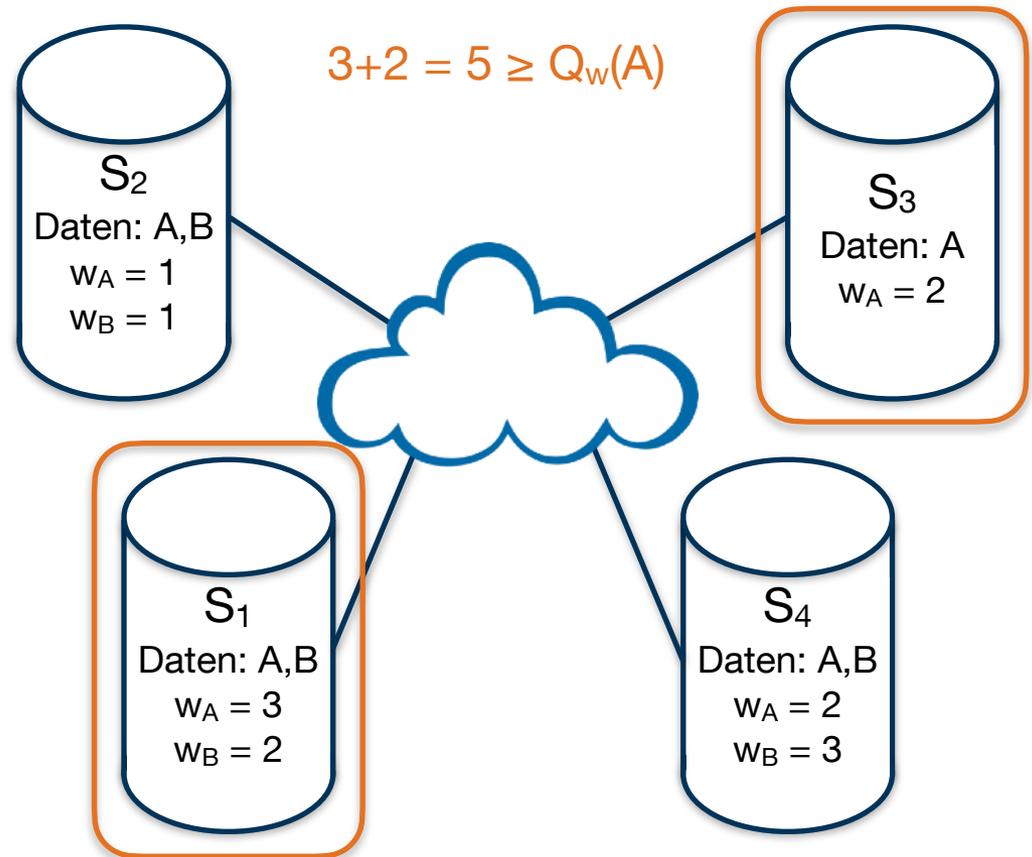
Verteilte Datenbanksysteme

Quorum-Consensus Verfahren

Schreibe:

- $A_- = 5$
- $B_+ = 2$

Station	Daten	Gewichte	Version
S ₁	A = 22 B = 7	w _A = 3 w _B = 2	A ₁ B ₁
S ₂	A = 22 B = 7	w _A = 1 w _B = 1	A ₁ B ₁
S ₃	A = 22	w _A = 2	A ₁
S ₄	A = 22 B = 7	w _A = 2 w _B = 3	A ₁ B ₁



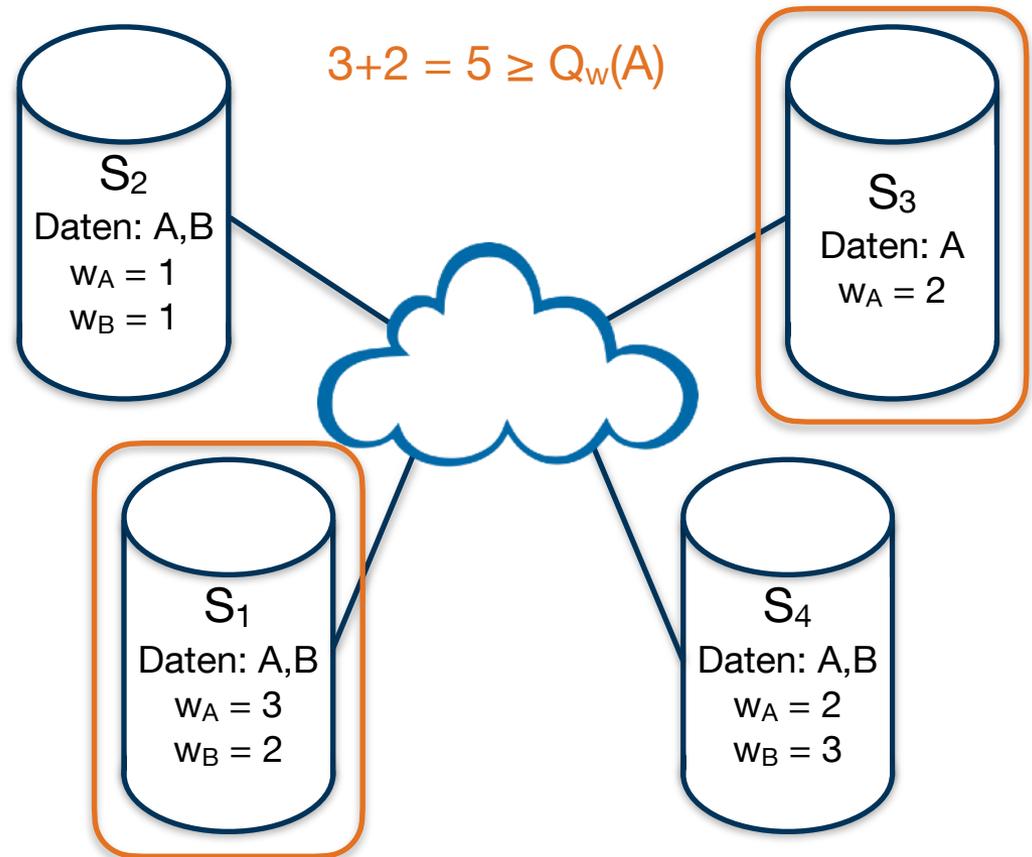
Verteilte Datenbanksysteme

Quorum-Consensus Verfahren

Schreibe:

- $A_- = 5$
- $B_+ = 2$

Station	Daten	Gewichte	Version
S ₁	A = 17 B = 7	w _A = 3 w _B = 2	A ₂ B ₁
S ₂	A = 22 B = 7	w _A = 1 w _B = 1	A ₁ B ₁
S ₃	A = 17	w _A = 2	A ₂
S ₄	A = 22 B = 7	w _A = 2 w _B = 3	A ₁ B ₁



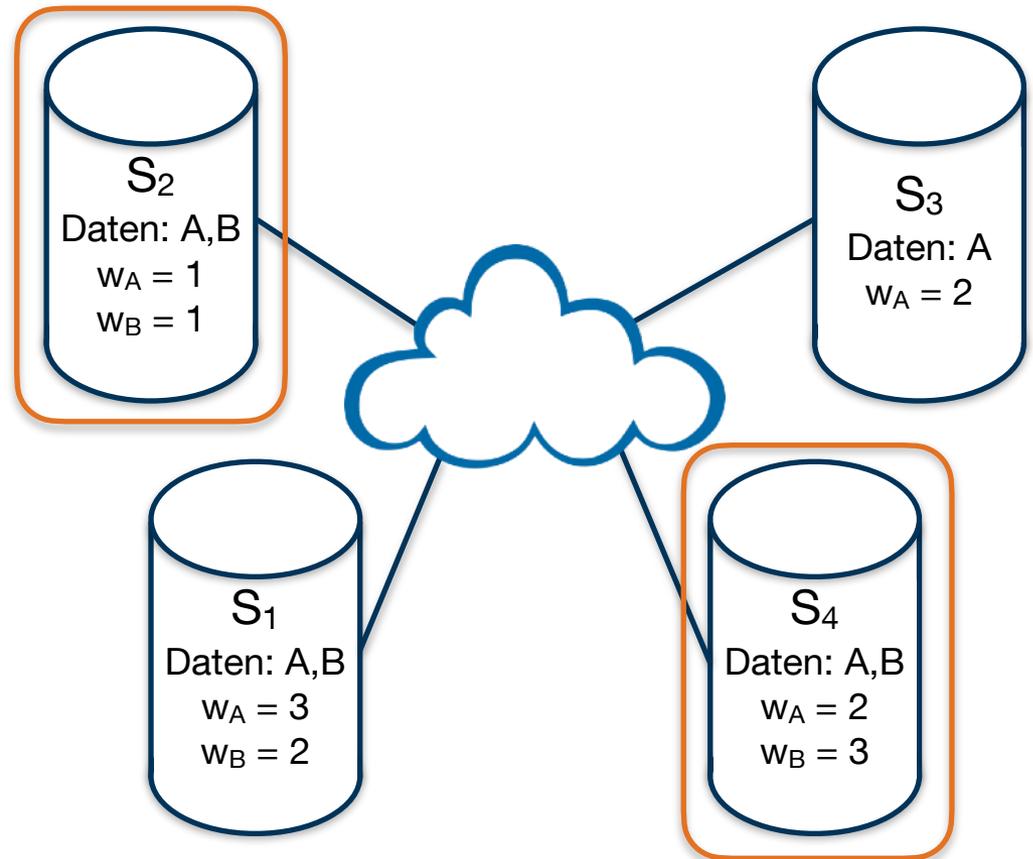
Verteilte Datenbanksysteme

Quorum-Consensus Verfahren

Schreibe:

- $A = 5$
- $B = 2$

Station	Daten	Gewichte	Version
S ₁	A = 17 B = 7	w _A = 3 w _B = 2	A ₂ B ₁
S ₂	A = 22 B = 7	w _A = 1 w _B = 1	A ₁ B ₁
S ₃	A = 17	w _A = 2	A ₂
S ₄	A = 22 B = 7	w _A = 2 w _B = 3	A ₁ B ₁



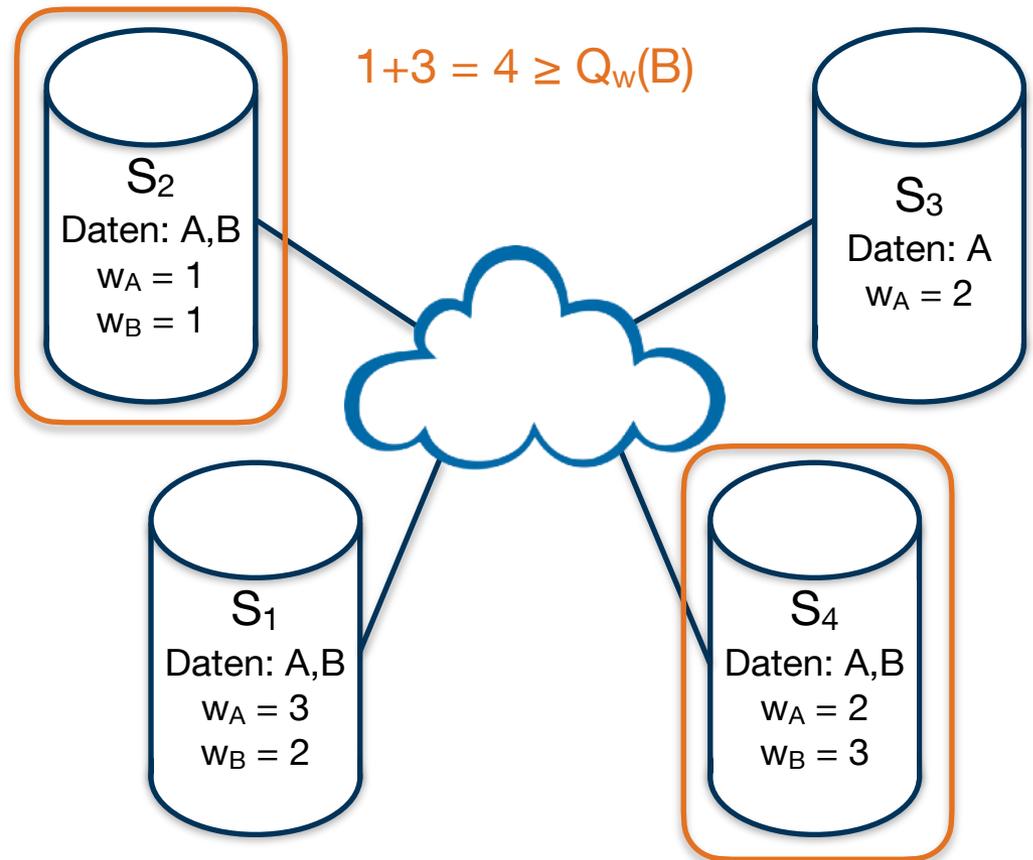
Verteilte Datenbanksysteme

Quorum-Consensus Verfahren

Schreibe:

- $A = 5$
- $B = 2$

Station	Daten	Gewichte	Version
S ₁	A = 17 B = 7	w _A = 3 w _B = 2	A ₂ B ₁
S ₂	A = 22 B = 7	w _A = 1 w _B = 1	A ₁ B ₁
S ₃	A = 17	w _A = 2	A ₂
S ₄	A = 22 B = 7	w _A = 2 w _B = 3	A ₁ B ₁



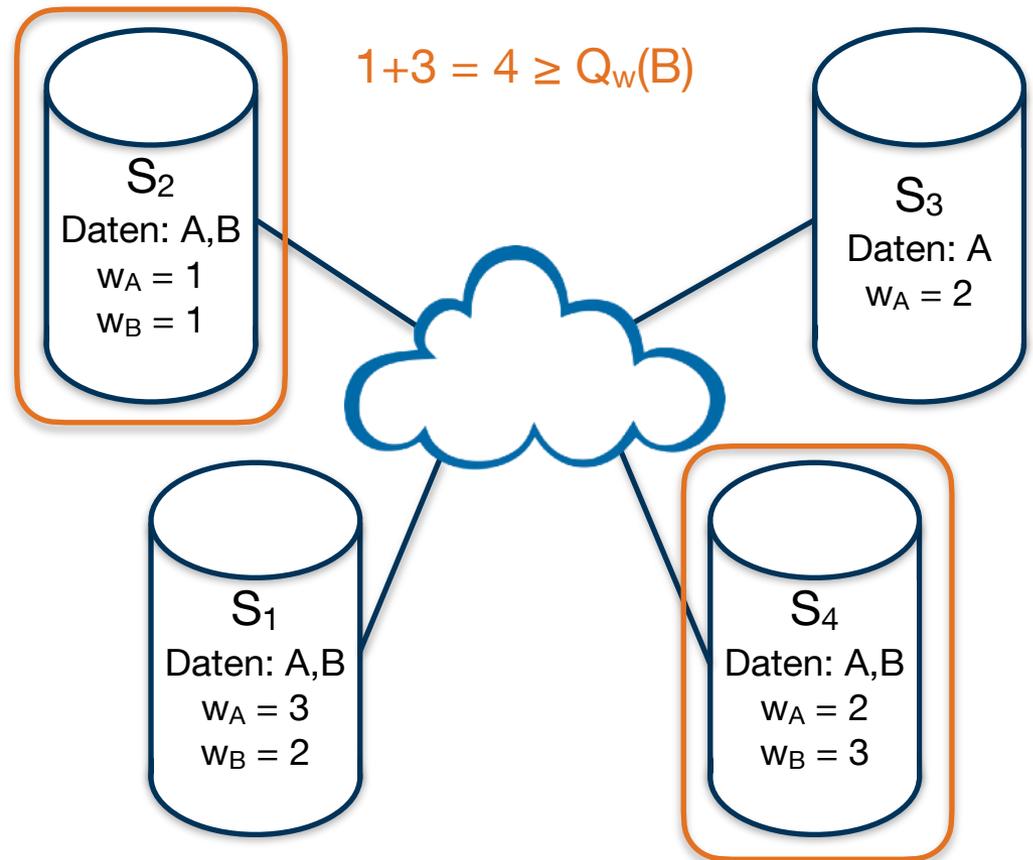
Verteilte Datenbanksysteme

Quorum-Consensus Verfahren

Schreibe:

- A- = 5
- B+ = 2

Station	Daten	Gewichte	Version
S ₁	A = 17 B = 7	w _A = 3 w _B = 2	A ₂ B ₁
S ₂	A = 22 B = 9	w _A = 1 w _B = 1	A ₁ B ₂
S ₃	A = 17	w _A = 2	A ₂
S ₄	A = 22 B = 9	w _A = 2 w _B = 3	A ₁ B ₂





Aufgabe 4

Zeigen Sie, dass die *write-all/read-any* Methode zur Synchronisation replizierter Daten einen Spezialfall der *Quorum-Consensus*-Methode darstellt.

- Für welche Art von Workloads eignet sich dieses Verfahren besonders gut?
- Wie werden Stimmen zugeordnet um *write-all/read-any* zu simulieren?
- Wie müssen die Quoren Q_w und Q_r vergeben werden?



Aufgabe 5

Um Ausfallsicherheit zu garantieren ist ein Datenwert 'A' auf vier Rechnern verteilt. Jeder Rechner hält dabei eine vollständige Kopie von 'A'. Um Konsistenz zu garantieren wird das Quorum-Consensus-Verfahren eingesetzt. Dabei ist jedem Rechner ein Gewicht $w_i(A)$ wie folgt zugewiesen:

Rechner	Kopie	Gewicht
R_1	A_1	3
R_2	A_2	1
R_3	A_3	2
R_4	A_4	2

Das Lesequorum ist $Q_r(A) = 4$ und das Schreibquorum ist $Q_w(A) = 5$.

- Geben Sie **alle** Lesemöglichkeiten für eine Transaktion auf dem Datum 'A' nach dem Quorum-Consensus-Protokoll an.
- Geben Sie **alle** Schreibmöglichkeiten für eine Transaktion auf dem Datum 'A' nach dem Quorum-Consensus-Protokoll an.
- Zeigen Sie für dieses Beispiel, dass während eine Transaktion T_1 ein Schreibquorum auf A hält es für andere Transaktionen T_x nicht möglich ist ein Lesequorum für A zu bekommen.

Verteilte Datenbanksysteme

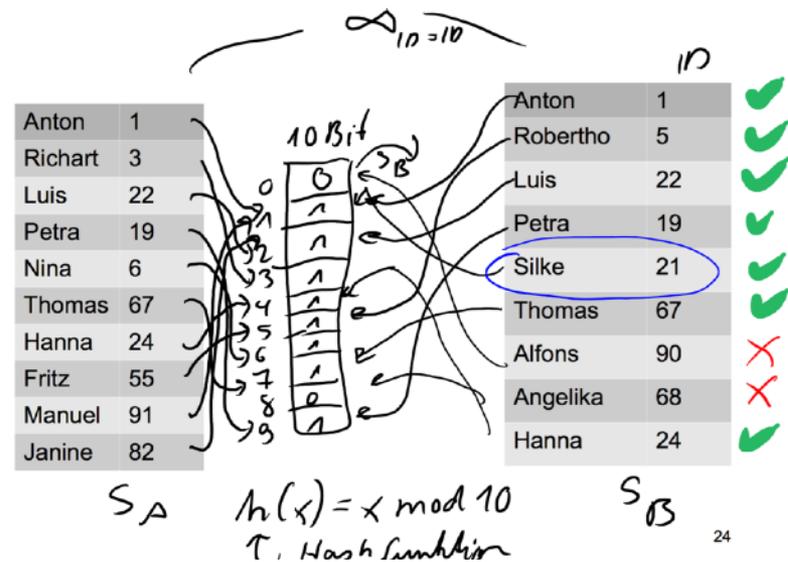
Bloom-Filter

- 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

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

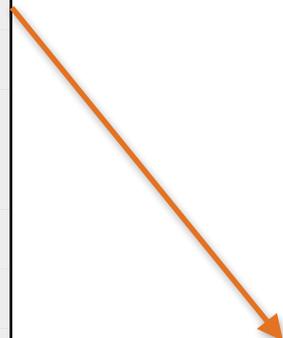
Verteilte Datenbanksysteme

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

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

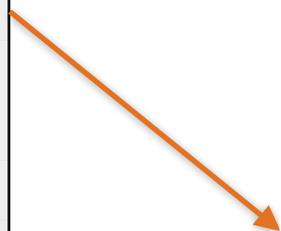
Verteilte Datenbanksysteme

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

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

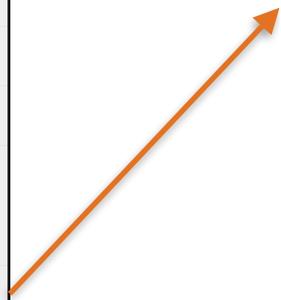
Verteilte Datenbanksysteme

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



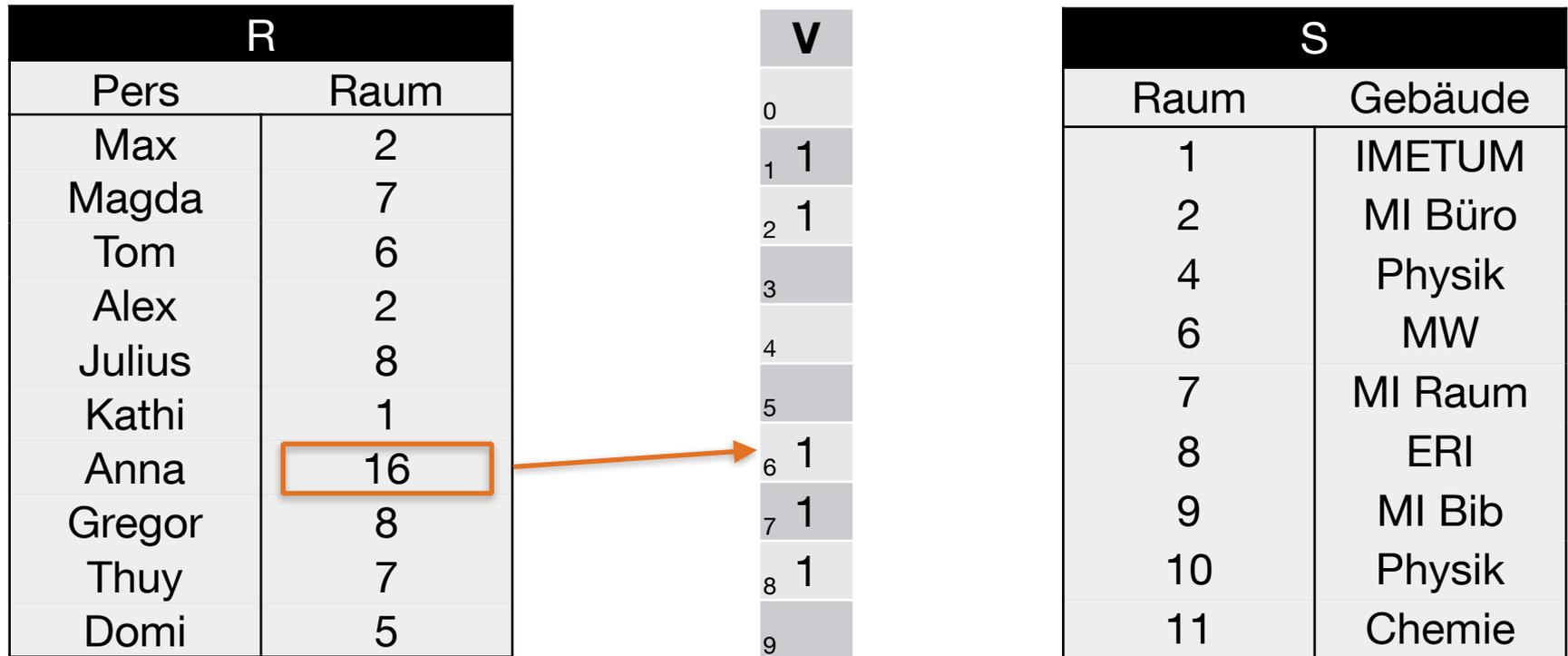
S	
Raum	Gebäude
1	IMETUM
2	MI Büro
4	Physik
6	MW
7	MI Raum
8	ERI
9	MI Bib
10	Physik
11	Chemie

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

Verteilte Datenbanksysteme

Bloom-Filter

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



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

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

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

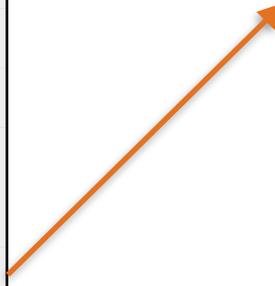
Verteilte Datenbanksysteme

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

$$h(x) = x \text{ mod } 10$$



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

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

False positive

- ✓ 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 6

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.



Fragen?