



# Einsatz und Realisierung von Datenbanksystemen

ERDB Übungsleitung

Alice Rey, Maximilian {Bandle, Schüle}, Michael Jungmair

[i3erdb@in.tum.de](mailto:i3erdb@in.tum.de)



# 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](mailto:i3erdb@in.tum.de) mit Angabe der Foliennummer.



# Big Data



# Big Data

## Term Frequency - Inverse Document Frequency

Anwendung im *Information Retrieval*

- Finde zu einer Suchanfrage die relevantesten Dokumente
- Große Herausforderung aufgrund der Menge an Web-Dokumenten

Term Frequency - Inverse Document Frequency (TF-IDF)

- Dokument-Ranking basierend auf Begriffshäufigkeiten
- Vollautomatische Analyse
- Meist wird nur ein *Vokabular* berücksichtigt, nicht alle Worte



# Big Data

## Term Frequency - Inverse Document Frequency

D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$TF_{ij} = \frac{f_{ij}}{\sum_{i=1 \dots |V|} f_{ij}}$$

#Wort i im Dokument j

#Worte im Dokument j

- Gewicht eines Begriffs in einem kurzen Dokument höher als in einem langen Dokument
- Normalisierung



# Big Data

## Term Frequency - Inverse Document Frequency

D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$TF_{ij} = \frac{f_{ij}}{\sum_{i=1 \dots |V|} f_{ij}}$$

#Wort i im Dokument j

#Worte im Dokument j

	TF		
Wort i	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
ERDB			
Klausur			
Erfolg			



# Big Data

## Term Frequency - Inverse Document Frequency

D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$TF_{ij} = \frac{f_{ij}}{\sum_{i=1 \dots |V|} f_{ij}}$$

#Wort i im Dokument j

#Worte im Dokument j

	TF		
Wort i	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
ERDB	1/5		
Klausur	0/5		
Erfolg	0/5		



# Big Data

## Term Frequency - Inverse Document Frequency

D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$TF_{ij} = \frac{f_{ij}}{\sum_{i=1 \dots |V|} f_{ij}}$$

#Wort i im Dokument j

#Worte im Dokument j

	TF		
Wort i	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
ERDB	1/5	0/5	
Klausur	0/5	1/5	
Erfolg	0/5	0/5	





# Big Data

## Term Frequency - Inverse Document Frequency

D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$TF_{ij} = \frac{f_{ij}}{\sum_{i=1 \dots |V|} f_{ij}}$$

#Wort i im Dokument j (points to  $f_{ij}$ )

#Worte im Dokument j (points to denominator)

	TF		
Wort i	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
ERDB	1/5	0/5	1/10
Klausur	0/5	1/5	1/10
Erfolg	0/5	0/5	1/10



# Big Data

## Term Frequency - Inverse Document Frequency

D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$IDF_i = \log(N/n_i)$$

#Dokumente

#Dokumente mit Wort i

- Gewichtung für jeden Begriff
- Seltene Begriffe bekommen eine höhere Gewichtung als Häufige



# Big Data

## Term Frequency - Inverse Document Frequency

D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$IDF_i = \log(N/n_i)$$

#Dokumente

#Dokumente mit Wort i

	IDF			
Wort <i>i</i>	N	n <sub>i</sub>	N/n <sub>i</sub>	log(N/n <sub>i</sub> )
ERDB				
Klausur				
Erfolg				



# Big Data

## Term Frequency - Inverse Document Frequency

D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$IDF_i = \log(N/n_i)$$

#Dokumente

#Dokumente mit Wort i

Wort <i>i</i>	IDF			
	N	n <sub>i</sub>	N/n <sub>i</sub>	log(N/n <sub>i</sub> )
<b>ERDB</b>	3			
<b>Klausur</b>	3			
<b>Erfolg</b>	3			



# Big Data

## Term Frequency - Inverse Document Frequency

D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$IDF_i = \log(N/n_i)$$

#Dokumente

#Dokumente mit Wort i

	IDF			
Wort <i>i</i>	N	n <sub>i</sub>	N/n <sub>i</sub>	log(N/n <sub>i</sub> )
<b>ERDB</b>	3	2		
<b>Klausur</b>	3	2		
<b>Erfolg</b>	3	1		

# Big Data

## Term Frequency - Inverse Document Frequency

D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$IDF_i = \log(N/n_i)$$

#Dokumente

#Dokumente mit Wort i

	IDF			
Wort <i>i</i>	N	n <sub>i</sub>	N/n <sub>i</sub>	log(N/n <sub>i</sub> )
<b>ERDB</b>	3	2	3/2	
<b>Klausur</b>	3	2	3/2	
<b>Erfolg</b>	3	1	3/1	



# Big Data

## Term Frequency - Inverse Document Frequency

D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

$$IDF_i = \log(N/n_i)$$

#Dokumente

#Dokumente mit Wort i

	IDF			
Wort <i>i</i>	N	n <sub>i</sub>	N/n <sub>i</sub>	log(N/n <sub>i</sub> )
<b>ERDB</b>	3	2	3/2	0,176
<b>Klausur</b>	3	2	3/2	0,176
<b>Erfolg</b>	3	1	3/1	0,477



# Big Data

## Term Frequency - Inverse Document Frequency

D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

Relevanz von Dokument j

$$rel(D_j, Q) = \sum_{i \in Q} TF_{ij} \times IDF_i$$

Anfrage Q

Für alle Wörter von Q





# Big Data

## Term Frequency - Inverse Document Frequency

D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

Relevanz von Dokument j

$$rel(D_j, Q) = \sum_{i \in Q} TF_{ij} \times IDF_i$$

Anfrage Q

Für alle Wörter von Q

$rel(D_1, \{ERDB, Klausur, Erfolg\}) =$

$$TF_{ERDB,1} * IDF_{ERDB} + TF_{Klausur,1} * IDF_{Klausur} + TF_{Erfolg,1} * IDF_{Erfolg}$$

# Big Data

## Term Frequency - Inverse Document Frequency

D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

Relevanz von Dokument j

$$rel(D_j, Q) = \sum_{i \in Q} TF_{ij} \times IDF_i$$

Anfrage Q

Für alle Wörter von Q

$$rel(D_1, \{ERDB, Klausur, Erfolg\}) =$$

$$TF_{ERDB,1} * IDF_{ERDB} + TF_{Klausur,1} * IDF_{Klausur} + TF_{Erfolg,1} * IDF_{Erfolg}$$

$$rel(D_1, \{ERDB, Klausur, Erfolg\}) =$$

$$1/5 * 0,176 + 0 * 0,176 + 0 * 0,477$$



# Big Data

## Term Frequency - Inverse Document Frequency

D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
ERDB macht echt viel Spaß	Die Klausur ist sicher machbar	Wir wünschen euch allen viel Erfolg bei der ERDB Klausur

Relevanz von Dokument j

$$rel(D_j, Q) = \sum_{i \in Q} TF_{ij} \times IDF_i$$

Anfrage Q

Für alle Wörter von Q

$$rel(D_1, \{ERDB, Klausur, Erfolg\}) =$$

$$TF_{ERDB,1} * IDF_{ERDB} + TF_{Klausur,1} * IDF_{Klausur} + TF_{Erfolg,1} * IDF_{Erfolg}$$

$$rel(D_1, \{ERDB, Klausur, Erfolg\}) =$$

$$1/5 * 0,176 + 0 * 0,176 + 0 * 0,477 = 0,0352$$



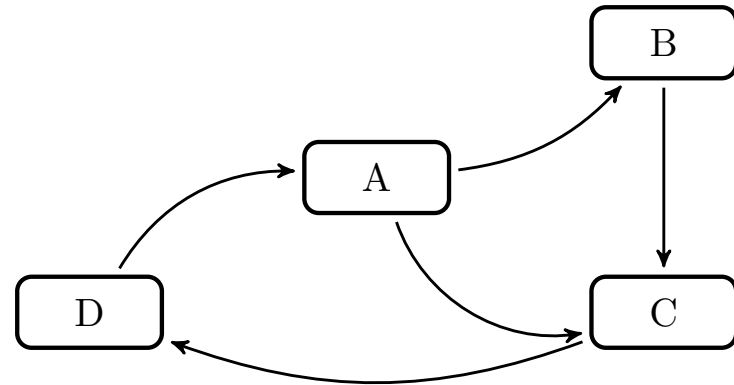
# Aufgabe 1

Berechnen Sie für folgende drei Dokumente die TF-IDF-Werte:

1. „Beim Fußball dauert ein Spiel neunzig Minuten – und am Ende gewinnen die Deutschen“
2. „Beim Fußball muss das Runde (der Ball) in das Eckige (das Tor)“
3. „Nie war ein Tor so wertvoll wie jetzt“

Welches Ranking ergibt sich gemäß der Relevanzwerte für die Anfrage: „Fußball“  $\wedge$  „Tor“. Zur Ermittlung des TF Wertes gehen sie davon aus, dass alle Wörter eines Dokuments *interessant* sind?

## Aufgabe 2



In dem in Abbildung 1 gezeigten Netzwerk von Web-Seiten wird ein kleines Beispiel für einen Webgraphen gezeigt. Lösen Sie folgende Aufgaben.

1. Berechnen Sie, für das in Abbildung gezeigte Netzwerk, den PageRank, sowie die HITS-Werte nach 2 Iterationen. Nutzen Sie  $1/|V|$  als Anfangswert für den PageRank und 1 für HITS.  $a = 0.1$
2. Formulieren sie eine Iteration des Pagerank Algorithmus in SQL. Der Graph ist dabei in der Tabelle  $edges(src, dst)$  gespeichert, die aktuelle PageRank Gewichtung in der Tabelle  $pagerank(node, pr)$ .
3. Formulieren Sie die SQL Anfrage nun als rekursive SQL Anfrage (100 Iterationen) um.



# HITS Algorithmus

## Hypertext Induced Topic Selection

Automatische Relevanz-Beurteilung für Websites  
Vernetzung als Kriterium

Zwei Rollen:

- Hub (Knotenpunkt)
  - Autorität (Website mit Inhalt)
- ➔ Alle Seiten werden in beiden Rollen beurteilt

Hub

- Wertvoller auf je mehr höherwertige Autoritäten er verweist (ausgehende Kanten)

Autorität

- Wertvoller je mehr höherwertige Hubs auf sie verweisen (eingehende Kanten)



# HITS Algorithmus

## Hypertext Induced Topic Selection

Iteration:

1. Berechne alle Hub-Werte

$$h_i = \sum_{j=1 \dots N} A_{ij} a_j$$

Summe der Gewichte der Knoten aller ausgehenden Kanten

2. Berechne alle Autoritäts-Werte

$$a_i = \sum_{j=1 \dots N} A_{ji} h_j$$

Summe der Gewichte der Knoten aller eingehenden Kanten

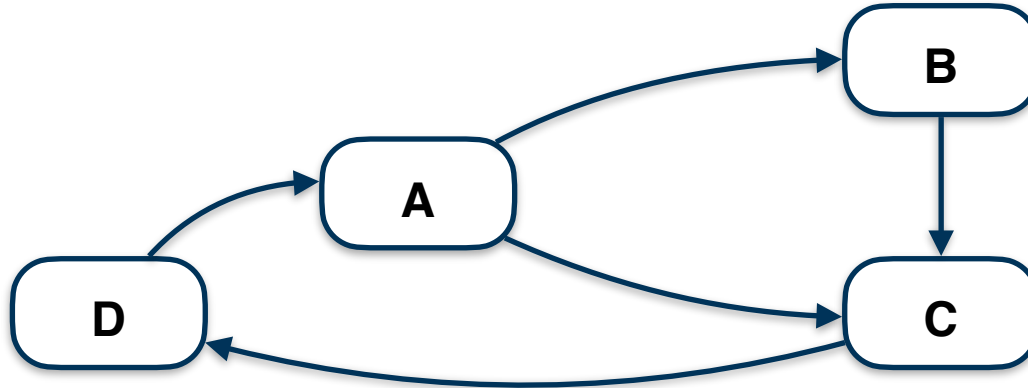
3. Normalisiere die Autoritäts-Werte mit

$$\lambda = \frac{1}{\max(a)}$$



## Aufgabe 2

### HITS Algorithmus



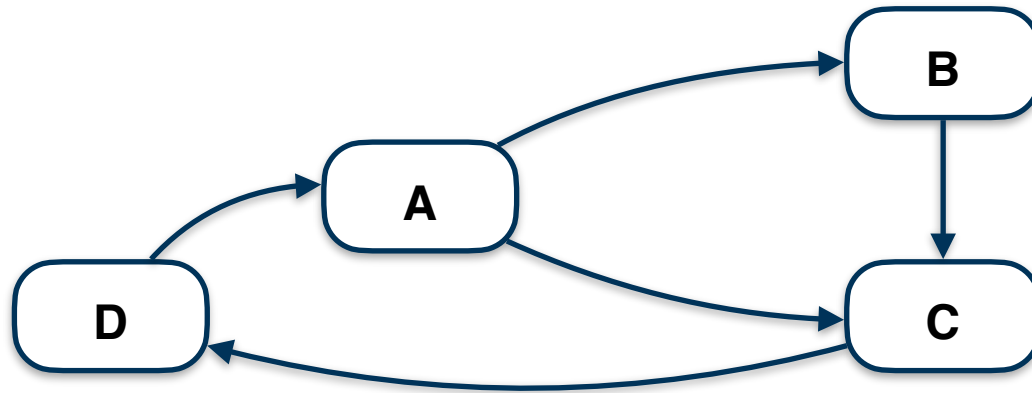
Berechne für den obigen Graphen die HITS-Werte nach 2 Iterationen.  
Nutze 1 als Startwert für HITS.





# Aufgabe 2

## HITS Algorithmus



1. Iteration

	A	B	C	D
Hubs	2	1	1	1
Vorläufige Autoritäten				
Normalisierte Autoritäten				

$$h_A = a_B + a_C = 1 + 1$$

$$h_B = a_C = 1$$

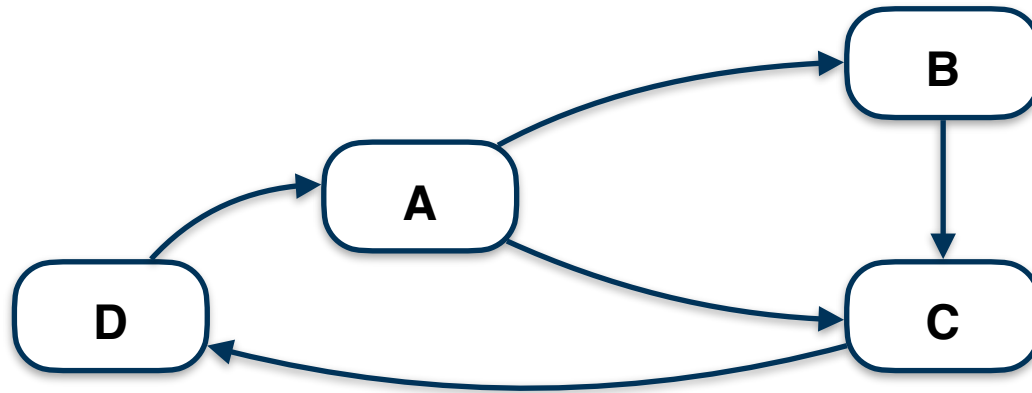
$$h_C = a_D = 1$$

$$h_D = a_A = 1$$



# Aufgabe 2

## HITS Algorithmus



1. Iteration

	A	B	C	D
Hubs	2	1	1	1
Vorläufige Autoritäten	1	2	3	1
Normalisierte Autoritäten				

$$a_A = h_D = 1$$

$$a_B = h_A = 2$$

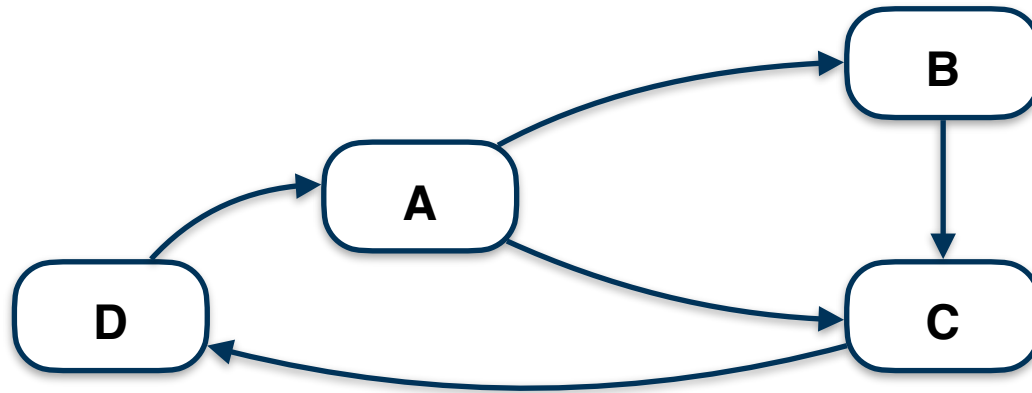
$$a_C = h_A + h_B = 2 + 1$$

$$a_D = h_C = 1$$



# Aufgabe 2

## HITS Algorithmus



1. Iteration

	A	B	C	D
Hubs	2	1	1	1
Vorläufige Autoritäten	1	2	3	1
Normalisierte Autoritäten	1/3	2/3	1	1/3

Normalisieren:

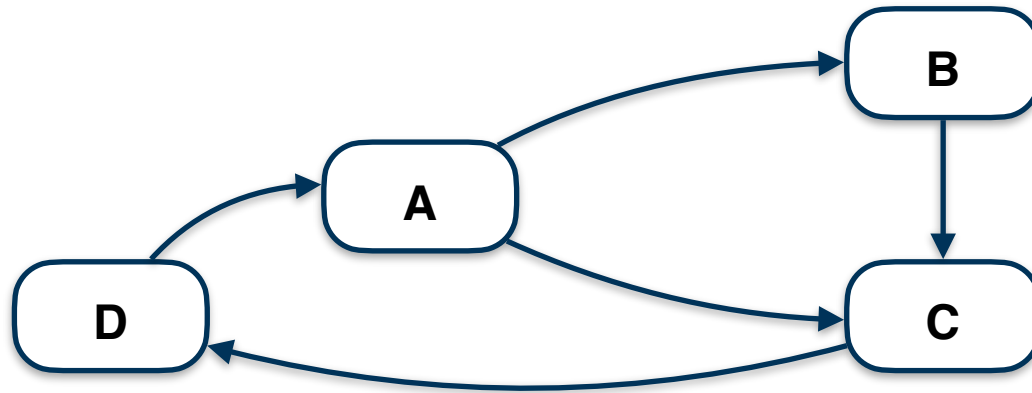
$$\max(a) = 3$$

$$\Rightarrow a_i * 1/3$$



# Aufgabe 2

## HITS Algorithmus



1. Iteration

	A	B	C	D
Hubs	2	1	1	1
Vorläufige Autoritäten	1	2	3	1
Normalisierte Autoritäten	1/3	2/3	1	1/3

2. Iteration

	A	B	C	D
Hubs	5/3	1	1/3	1/3
Vorläufige Autoritäten				
Normalisierte Autoritäten				

$$h_A = a_B + a_C = 2/3 + 1$$

$$h_B = a_C = 1$$

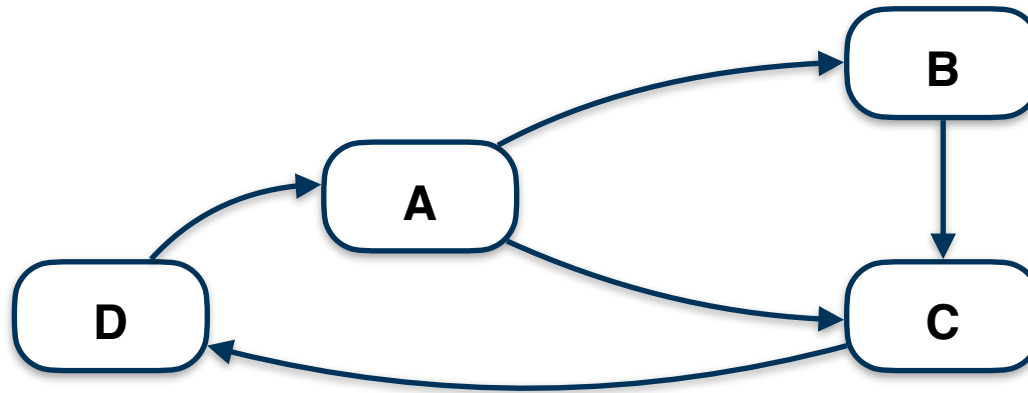
$$h_C = a_D = 1/3$$

$$h_D = a_A = 1/3$$



# Aufgabe 2

## HITS Algorithmus



1. Iteration

	A	B	C	D
Hubs	2	1	1	1
Vorläufige Autoritäten	1	2	3	1
Normalisierte Autoritäten	1/3	2/3	1	1/3

2. Iteration

	A	B	C	D
Hubs	5/3	1	1/3	1/3
Vorläufige Autoritäten	1/3	5/3	8/3	1/3
Normalisierte Autoritäten				

$$a_A = h_D = 1/3$$

$$a_B = h_A = 5/3$$

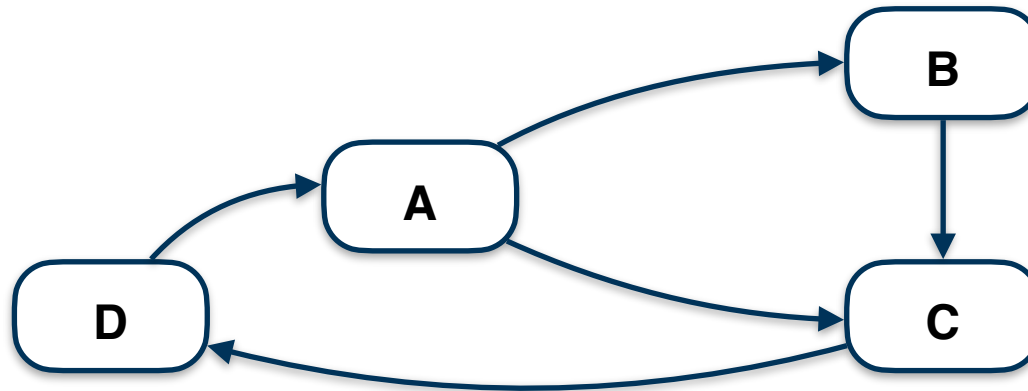
$$a_C = h_A + h_B = 5/3 + 1$$

$$a_D = h_C = 1/3$$



# Aufgabe 2

## HITS Algorithmus



1. Iteration

	A	B	C	D
Hubs	2	1	1	1
Vorläufige Autoritäten	1	2	3	1
Normalisierte Autoritäten	1/3	2/3	1	1/3

2. Iteration

	A	B	C	D
Hubs	5/3	1	1/3	1/3
Vorläufige Autoritäten	1/3	5/3	8/3	1/3
Normalisierte Autoritäten	1/8	5/8	1	1/8

Normalisieren:

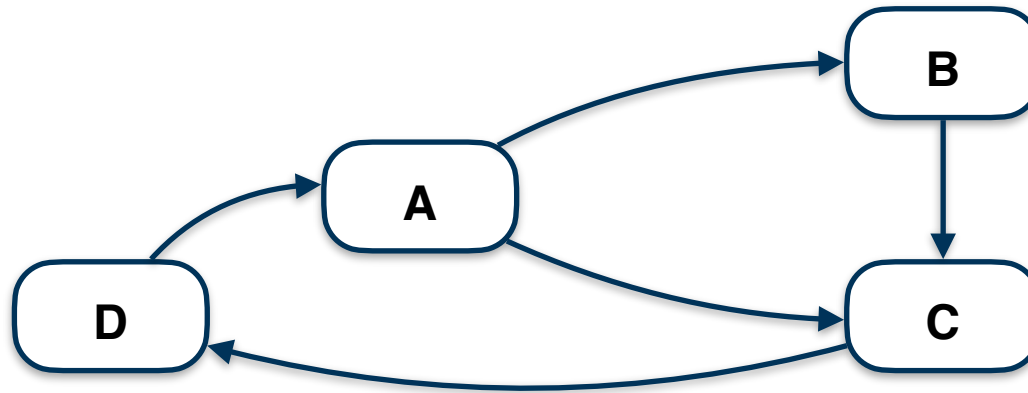
$$\max(a) = 8/3$$

$$\Rightarrow a_i * 3/8$$



# Aufgabe 2

## HITS Algorithmus



1. Iteration

	A	B	C	D
Hubs	2	1	1	1
Vorläufige Autoritäten	1	2	3	1
Normalisierte Autoritäten	1/3	2/3	1	1/3

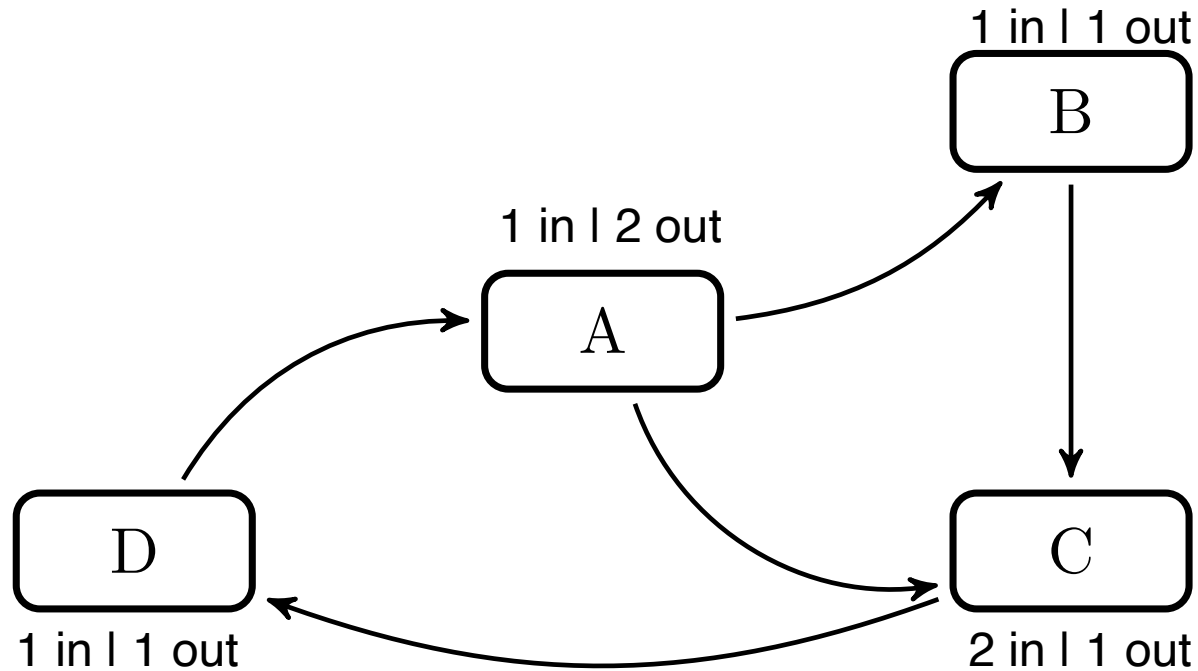
2. Iteration

	A	B	C	D
Hubs	5/3	1	1/3	1/3
Vorläufige Autoritäten	1/3	5/3	8/3	1/3
Normalisierte Autoritäten	1/8	5/8	1	1/8



## Aufgabe 2

### Pagerank - Berechnung am Graphen







## Aufgabe 2

### Pagerank

```
2. select VTo, 0.1/(CAST((select count(*) from pagerank)AS FLOAT))
    +0.9*sum(Beitrag)
from(
    select e.VTo, p.Weight/
        (select count(*) from edges x where x.VFrom=e.VFrom) as Beitrag
    from edges e , pagerank p
    where e.VFrom=p.Vertex
) i
group by VTo
```



# Zentralitätsmaße

- Aus den Sozialwissenschaften
- Charakterisiert ganze Graphen oder Teilstrukturen
- Sinnhaftigkeit ist manchmal zweifelhaft

# Zentralitätsmaße

## Verbindungszentralität

Verbindungszentralität

$$C_D(G) = \sum_{i=1}^{|V|} [C_D(v^*) - C_D(v_i)]$$

$v^*$  = Knoten mit max. Grad

Für Sternförmige Graphen, wird zur Normierung genutzt

$$C_D(G^*) = \sum_{i=1}^{|V|} [C_D(v^*) - C_D(v_i)] = (|V| - 2)(|V| - 1)$$

Normierte Verbindungszentralität

$$C'_D(G) = C_D(G) / [(|V| - 2)(|V| - 1)]$$



# Zentralitätsmaße

## Nähe-Zentralität

$$H(v) = \sum_{y \in V, y \neq v} 1/d(y, v)$$

Hierbei definiert man  $1/\infty$  als 0.

# Zentralitätsmaße

## Pfad-Zentralität

Für einen Knoten  $v$  im Graph  $G = (V, E)$  wird also dieser Wert wie folgt bestimmt:

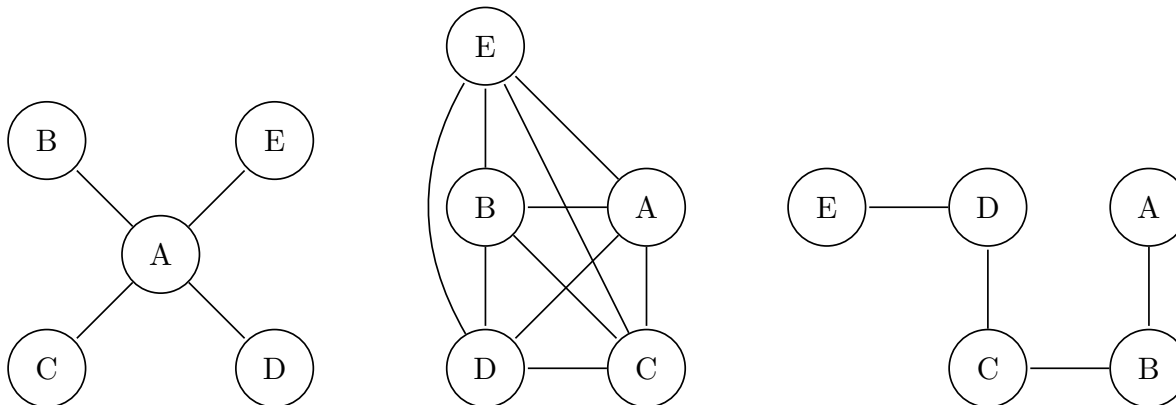
1. Für jedes Knotenpaar  $(s, t)$  berechne deren kürzeste Pfade im Graphen.
2. Bestimme die Anzahl der kürzesten Pfade von  $s$  nach  $t$  als  $\sigma_{st}$ .
3. Für jedes Knotenpaar  $(s, t)$  bestimme die Anzahl der kürzesten Pfade, die durch den betrachteten Knoten  $v$  verlaufen. Dieser Wert sei als  $\sigma_{st}(v)$  bezeichnet.

$$C_B(v) = \sum_{s \neq v \neq t \in V} (\sigma_{st}(v) / \sigma_{st})$$

## Aufgabe 3

In Abbildung 2 sind drei Graphen gegeben, ein sternförmiger, eine Clique und ein linear angeordneter.

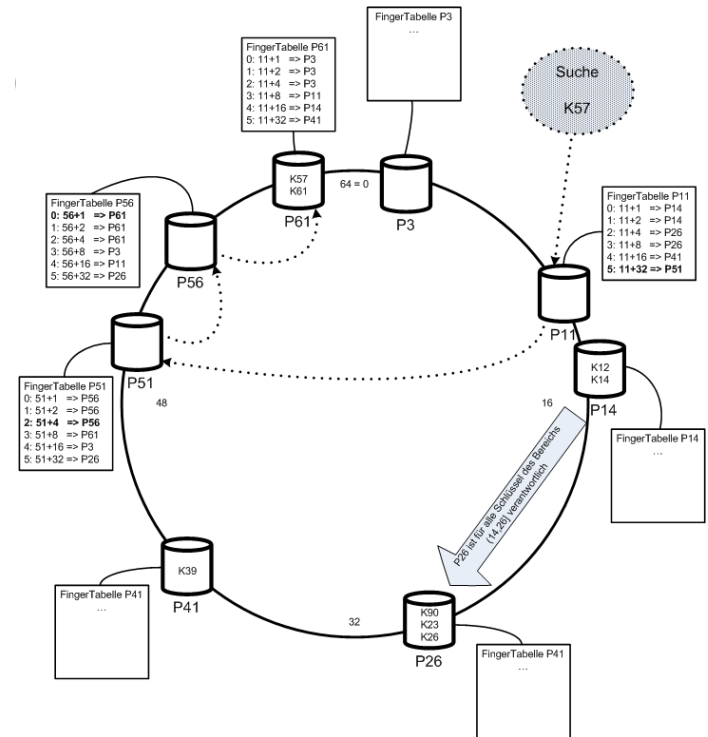
1. Berechnen Sie den Grad der Knoten für jeden der Graphen.
2. Berechnen Sie die Verbindungszentralität  $C_D(G)$  der drei Graphen, sowie deren normierte Verbindungszentralität  $C'_D(G)$ .
3. Berechnen Sie die Nähe-Zentralität  $H_G(v)$  für einen Knoten der drei Graphen.
4. Berechnen Sie die Pfad-Zentralität  $H_G(v)$  für einen Knoten der drei Graphen.



# Verteilte Datenbanksysteme

## Chord-Overlaynetzwerk

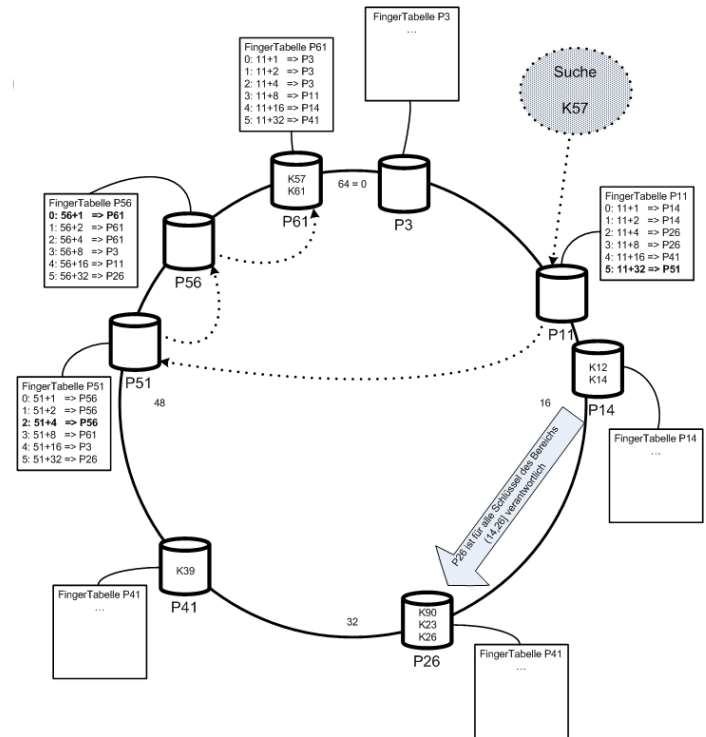
- Dezentral organisiert
- Alle Peers haben die gleiche Funktionalität
- Anfragen werden zielgerichtet geroutet
- Peers werden mittels Hashfunktion auf Zahlenring  $[0 \dots 2^n)$  platziert (IP)
- Datenobjekte werden auf den Zahlenring gehasht (durch Suchschlüssel)



# Verteilte Datenbanksysteme

## Chord-Overlaynetzwerk

- Peer ist für Daten zwischen Vorgänger und seiner Hashzahl verantwortlich
- Fällt ein Peer aus, wird der Bereich dem Nachfolger zugeordnet
- Jeder Peer hat eine FingerTabelle:
  - Enthält IP-Adressen von  $\log(\#\text{Peers})$  Peers

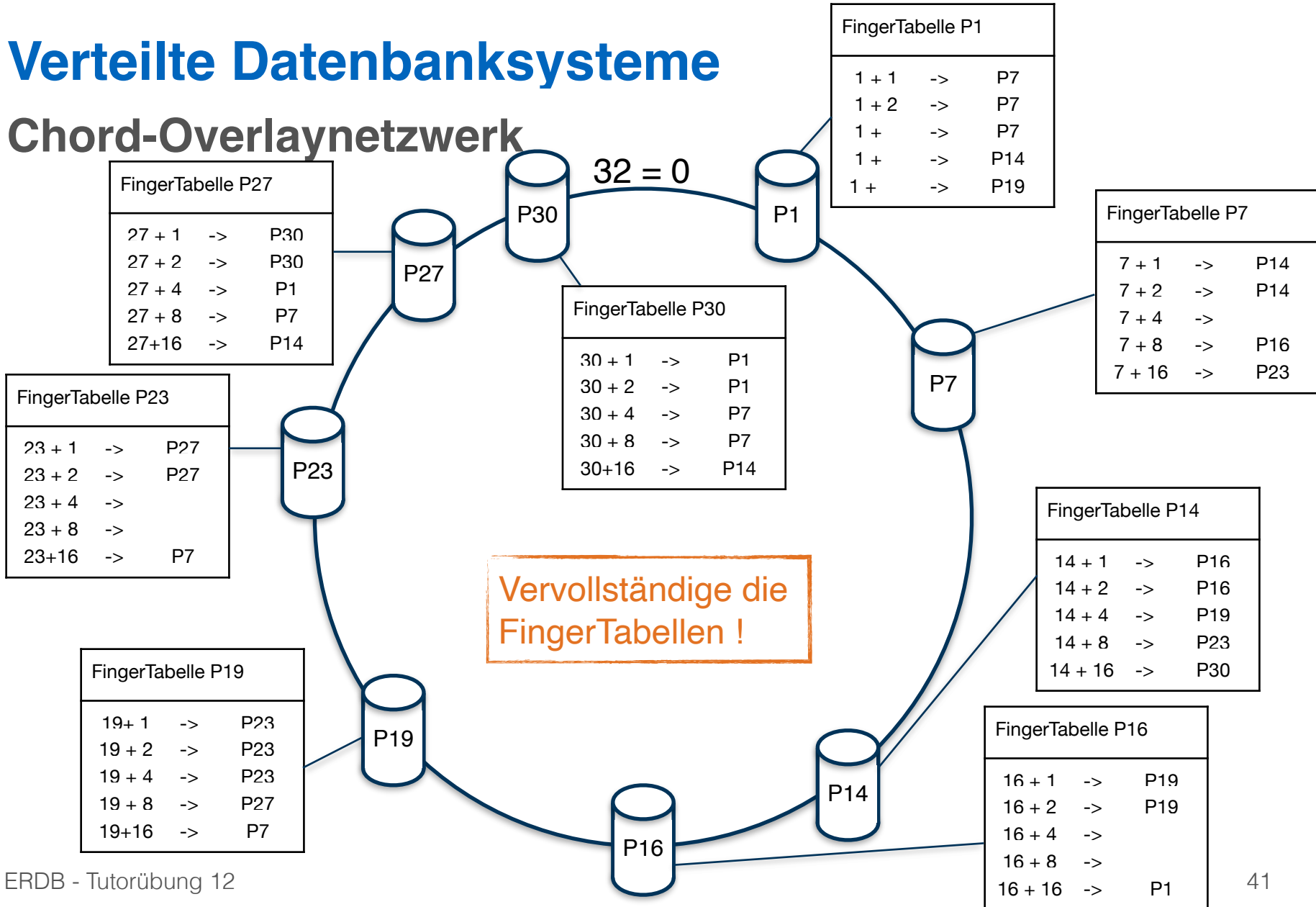






# Verteilte Datenbanksysteme

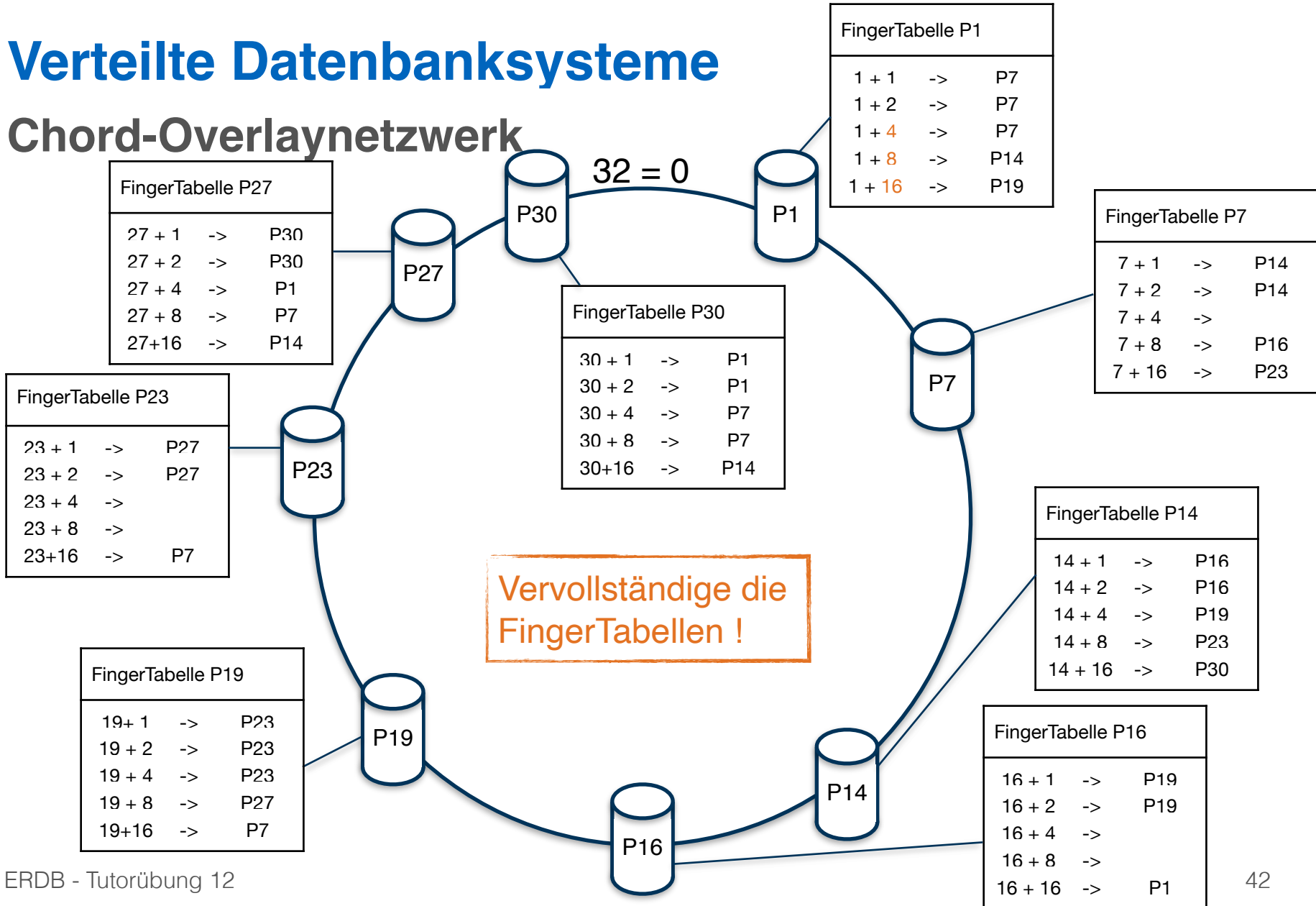
## Chord-Overlaynetzwerk





# Verteilte Datenbanksysteme

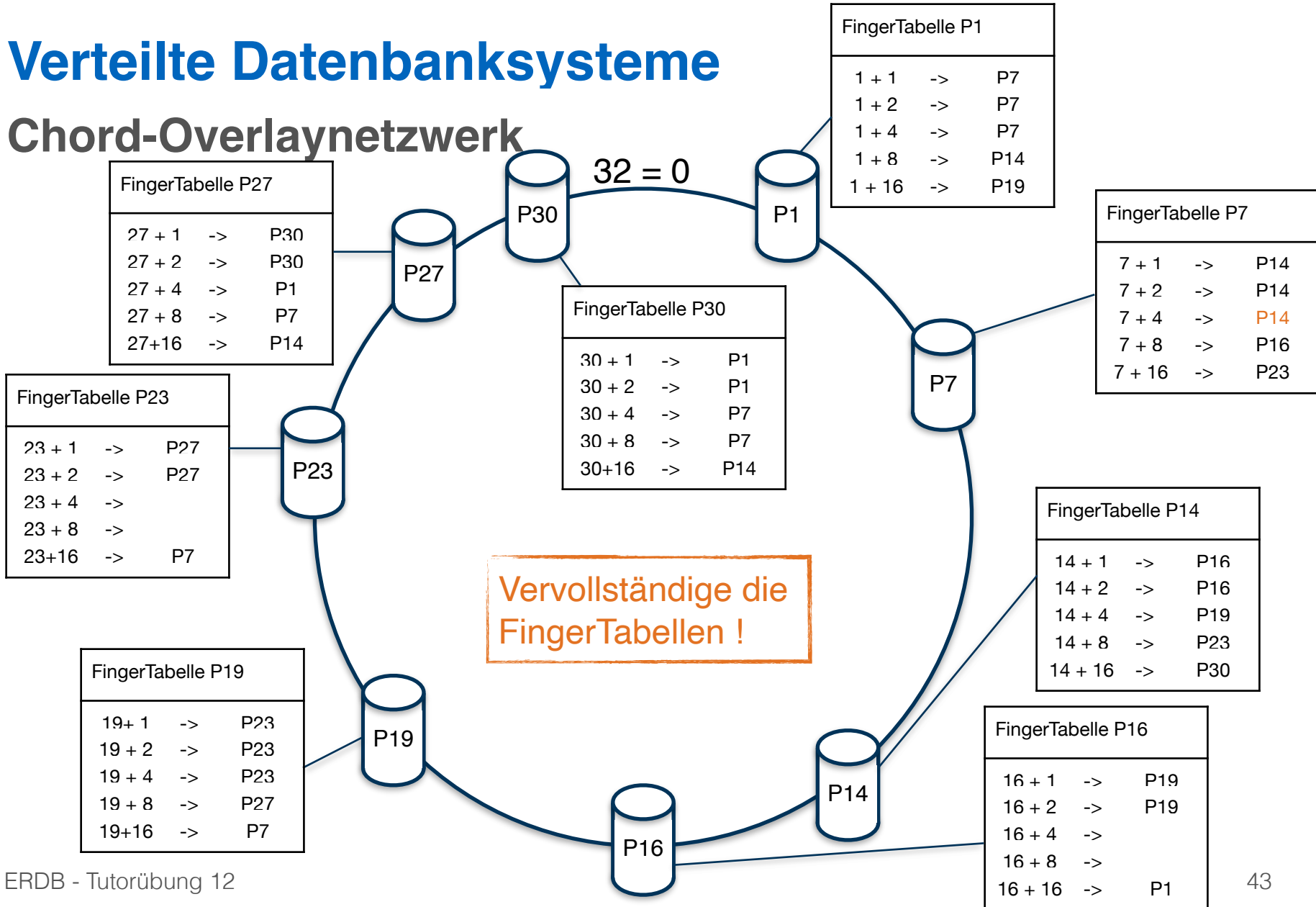
## Chord-Overlaynetzwerk





# Verteilte Datenbanksysteme

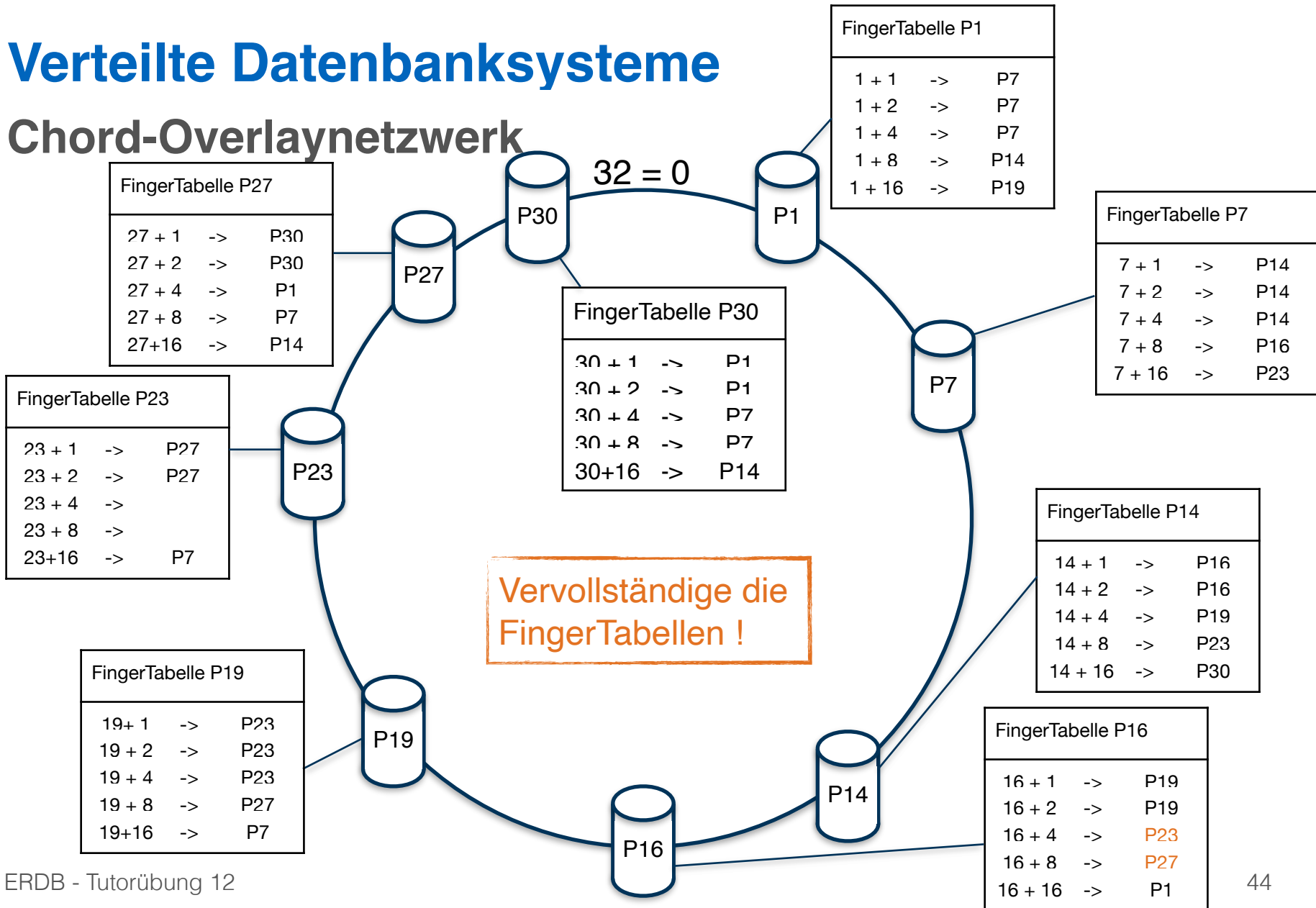
## Chord-Overlaynetzwerk





# Verteilte Datenbanksysteme

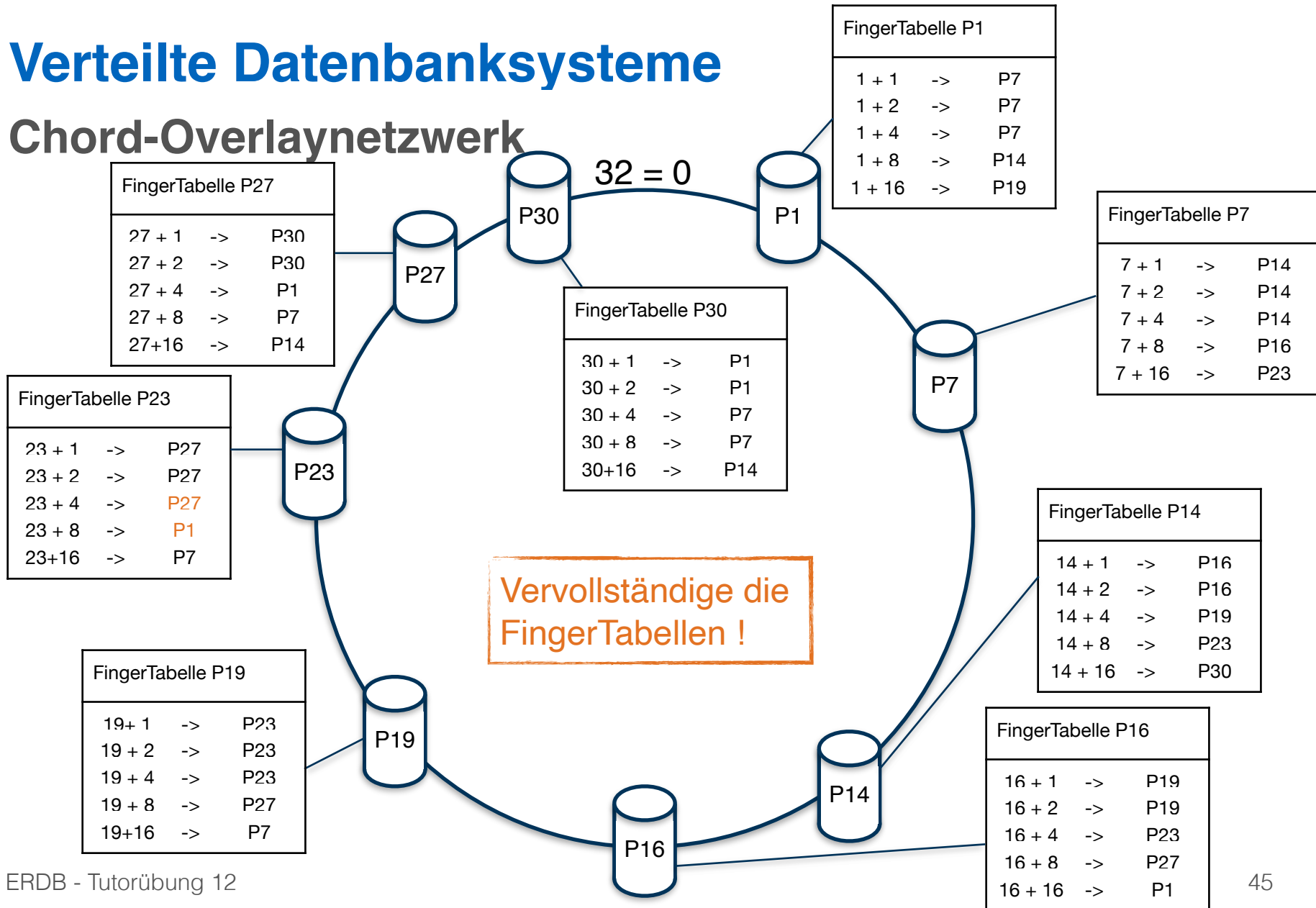
## Chord-Overlaynetzwerk





# Verteilte Datenbanksysteme

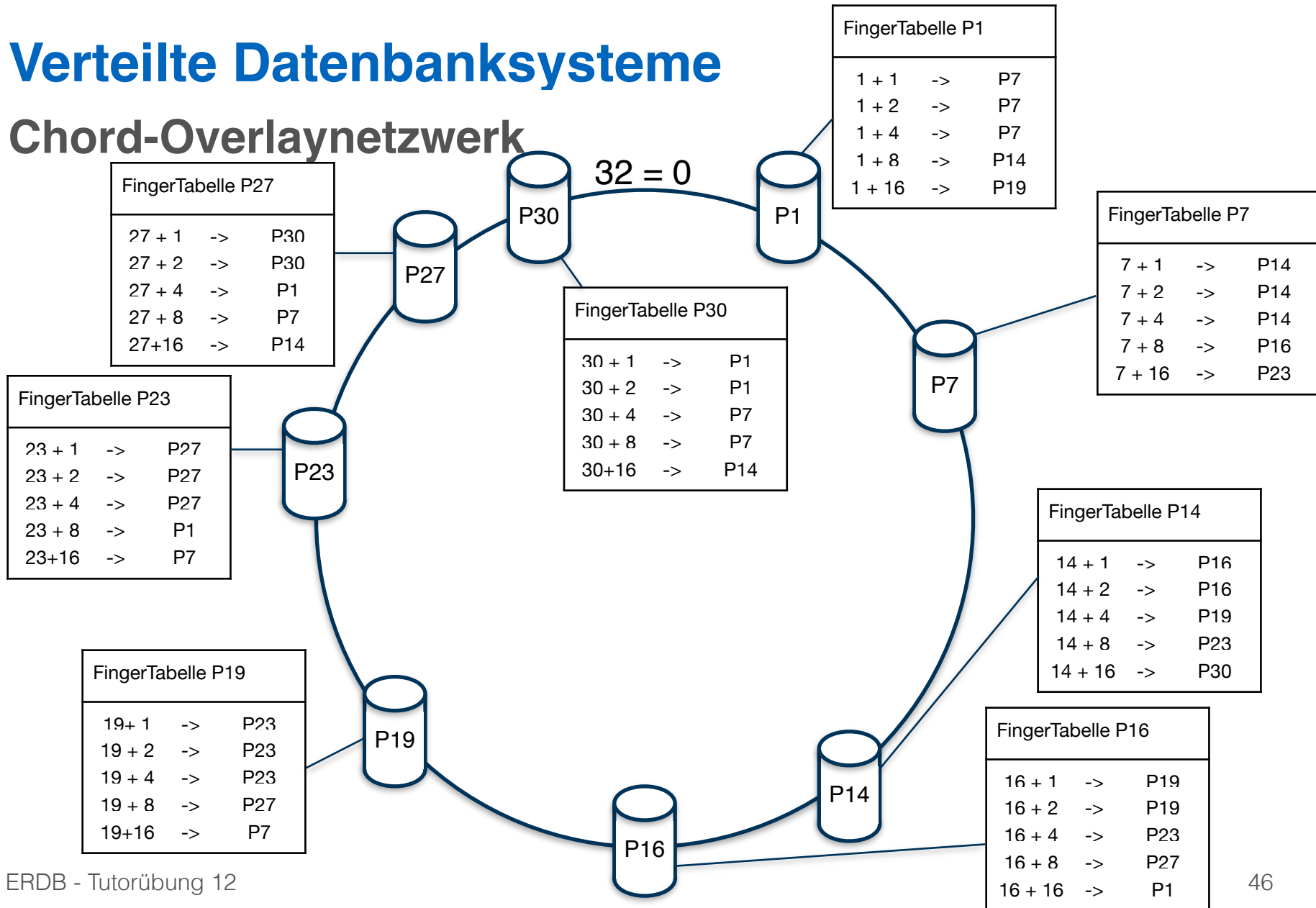
## Chord-Overlaynetzwerk





# Verteilte Datenbanksysteme

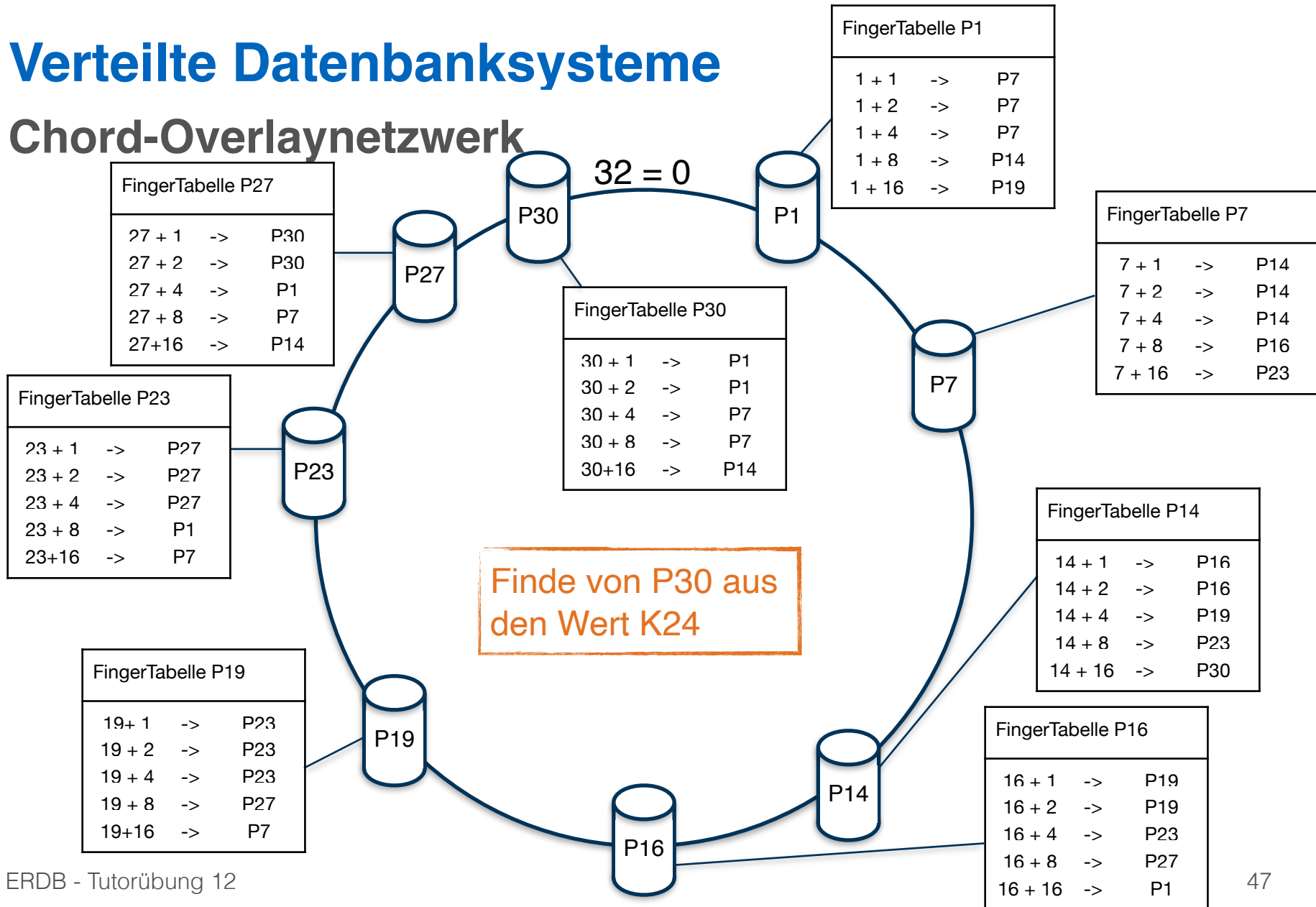
## Chord-Overlaynetzwerk





# Verteilte Datenbanksysteme

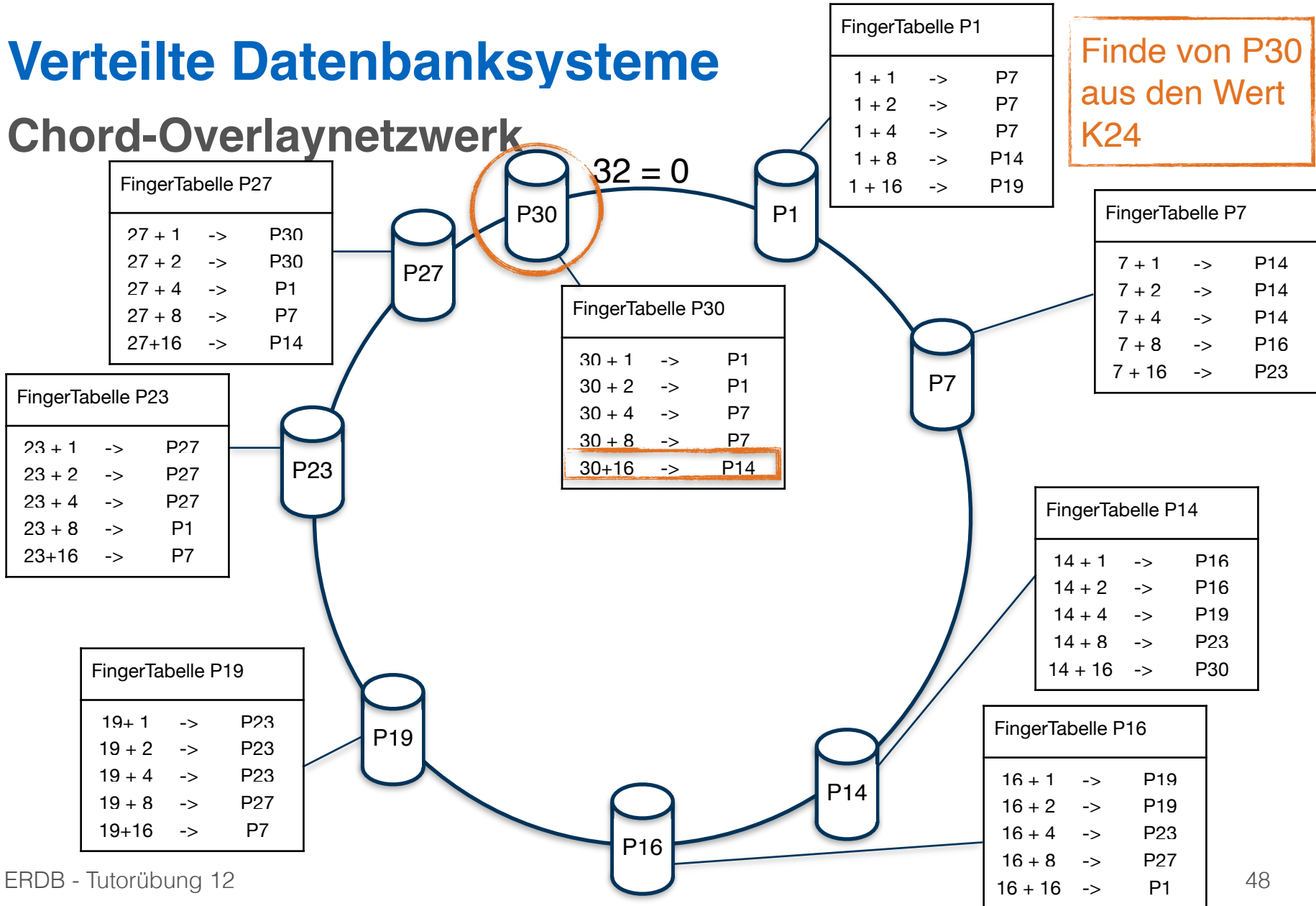
## Chord-Overlaynetzwerk





# Verteilte Datenbanksysteme

## Chord-Overlaynetzwerk

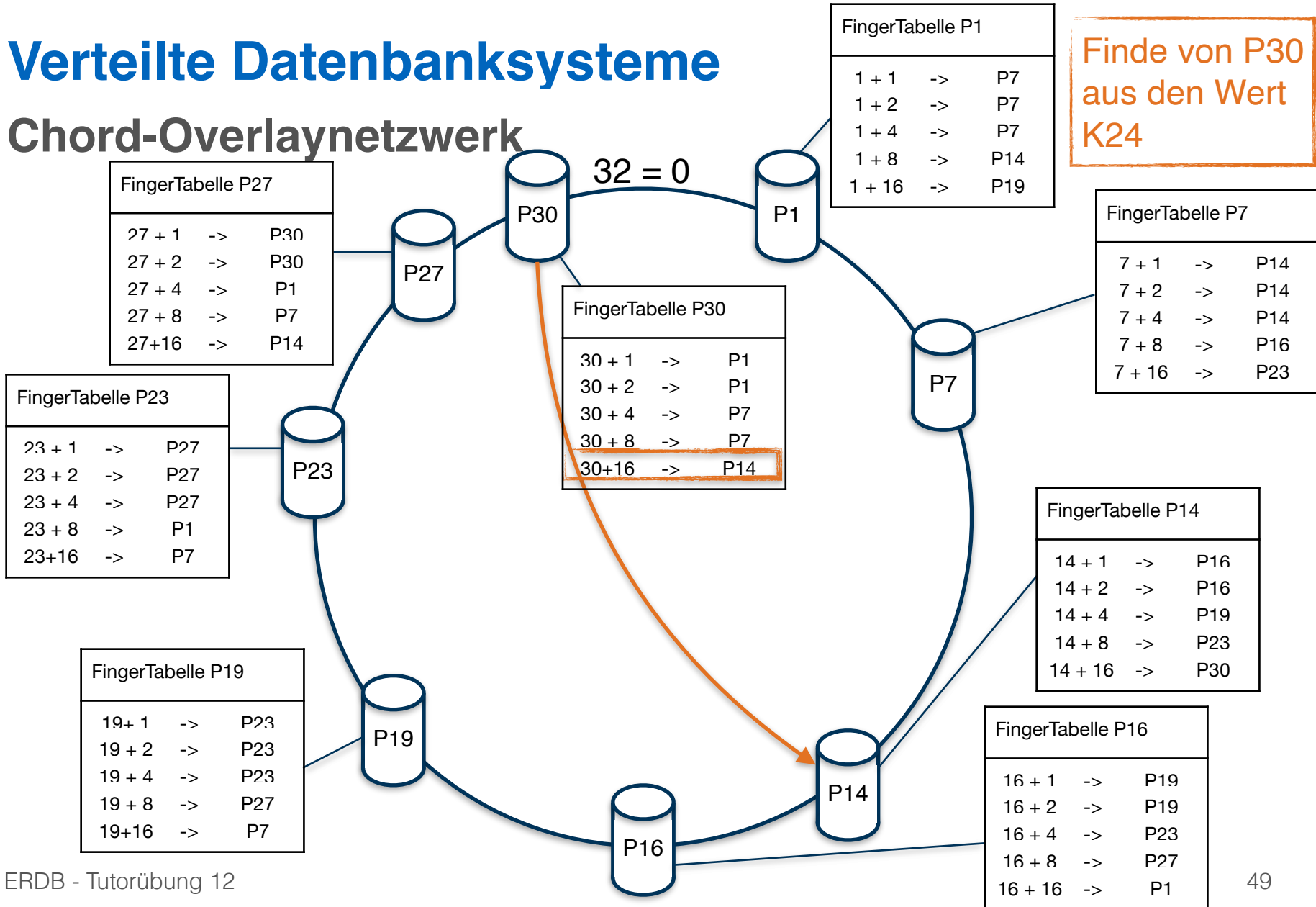






# Verteilte Datenbanksysteme

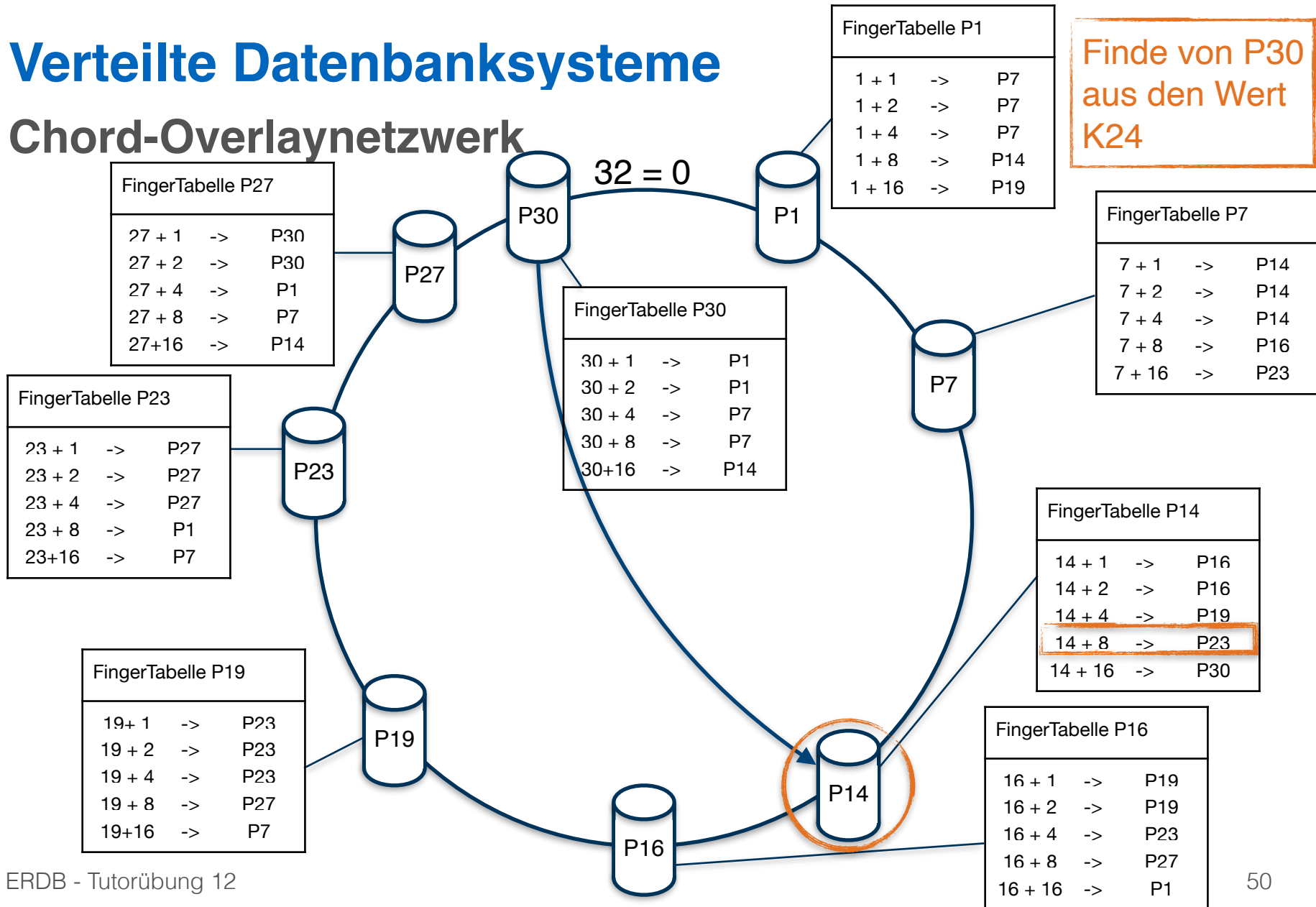
## Chord-Overlaynetzwerk





# Verteilte Datenbanksysteme

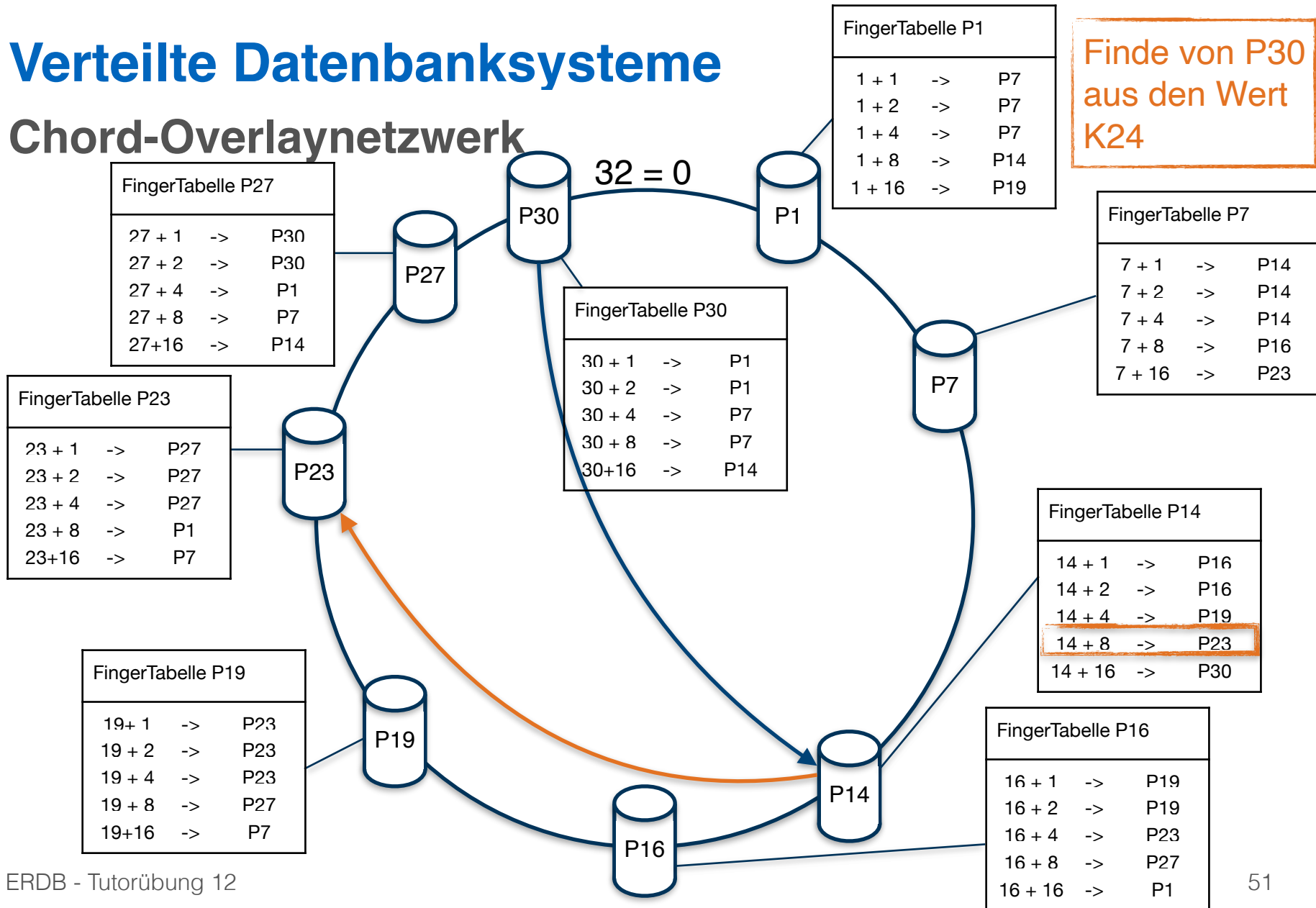
## Chord-Overlaynetzwerk





# Verteilte Datenbanksysteme

## Chord-Overlaynetzwerk





# Verteilte Datenbanksysteme

## Chord-Overlaynetzwerk

Finde von P30  
aus den Wert  
K24

FingerTabelle P27		
27 + 1	->	P30
27 + 2	->	P30
27 + 4	->	P1
27 + 8	->	P7
27+16	->	P14

FingerTabelle P23		
23 + 1	->	P27
23 + 2	->	P27
23 + 4	->	P27
23 + 8	->	P1
23+16	->	P7

FingerTabelle P19		
19+ 1	->	P23
19 + 2	->	P23
19 + 4	->	P23
19 + 8	->	P27
19+16	->	P7

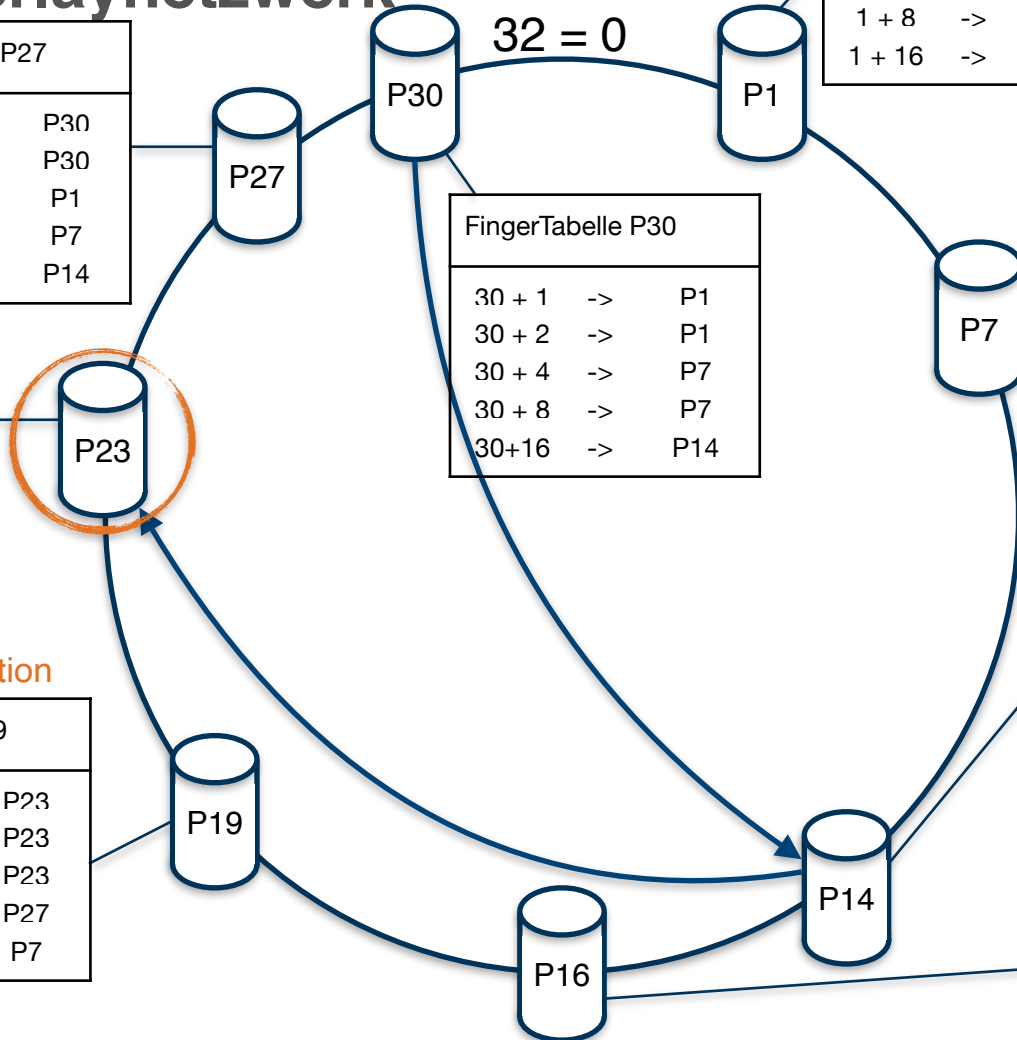
FingerTabelle P30		
30 + 1	->	P1
30 + 2	->	P1
30 + 4	->	P7
30 + 8	->	P7
30+16	->	P14

FingerTabelle P1		
1 + 1	->	P7
1 + 2	->	P7
1 + 4	->	P7
1 + 8	->	P14
1 + 16	->	P19

FingerTabelle P7		
7 + 1	->	P14
7 + 2	->	P14
7 + 4	->	P14
7 + 8	->	P16
7 + 16	->	P23

FingerTabelle P14		
14 + 1	->	P16
14 + 2	->	P16
14 + 4	->	P19
14 + 8	->	P23
14 + 16	->	P30

FingerTabelle P16		
16 + 1	->	P19
16 + 2	->	P19
16 + 4	->	P23
16 + 8	->	P27
16 + 16	->	P1

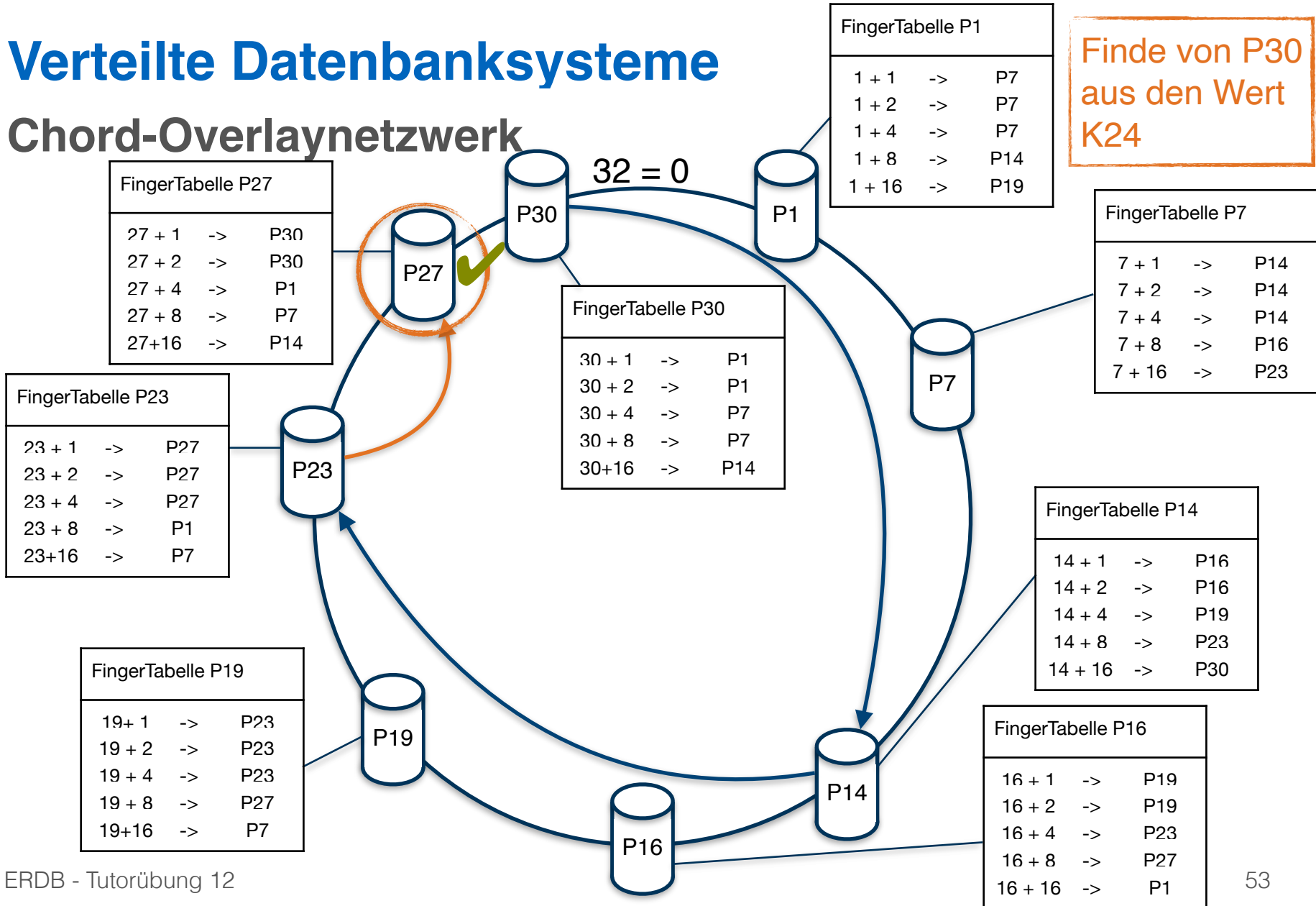


alle Stationen  $\geq K24$   
 $\Rightarrow K24$  von nächster Station  
verwaltet



# Verteilte Datenbanksysteme

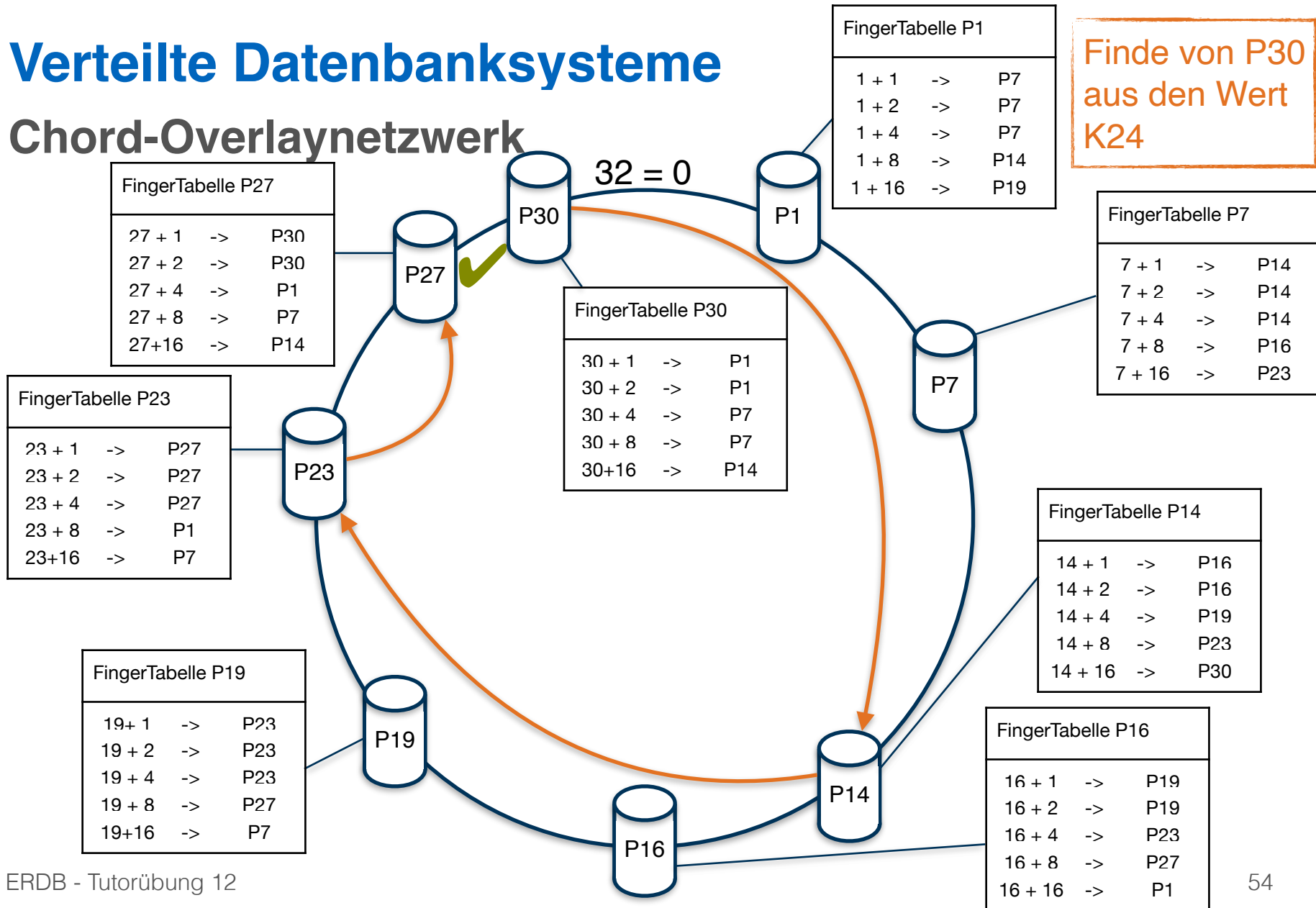
## Chord-Overlaynetzwerk





# Verteilte Datenbanksysteme

## Chord-Overlaynetzwerk



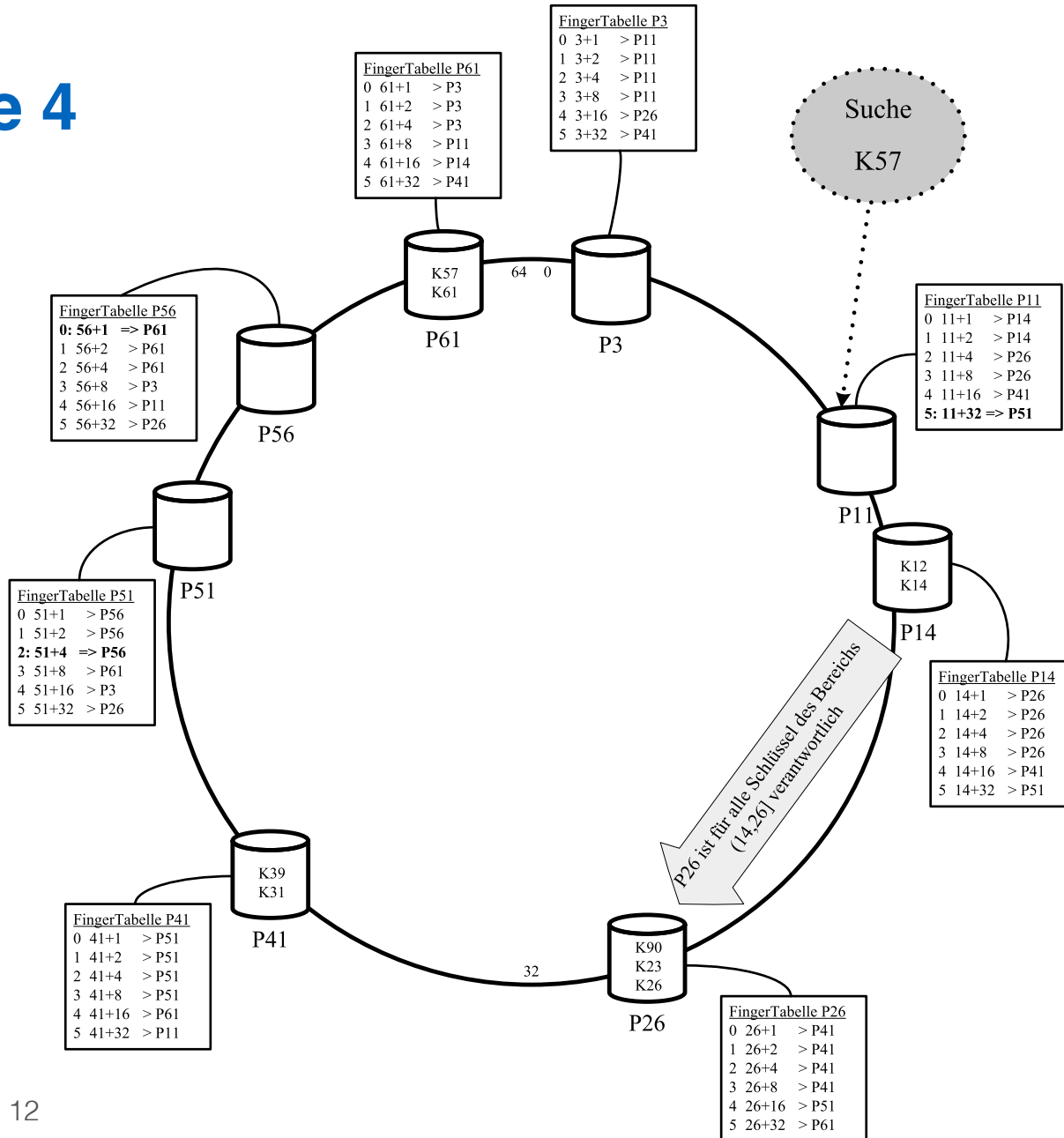


## Aufgabe 4

Zeigen Sie, dass die Suche in einem Chord-Overlaynetzwerk durch die Nutzung der FingerTabellen in maximal logarithmisch vielen Schritten zur Größe des Zahlenrings (bzw. der Anzahl der Stationen) durchgeführt werden kann. Verwenden Sie die Suche nach K57 beginnend an Station P11 (siehe Abbildung 2) zur Illustration.



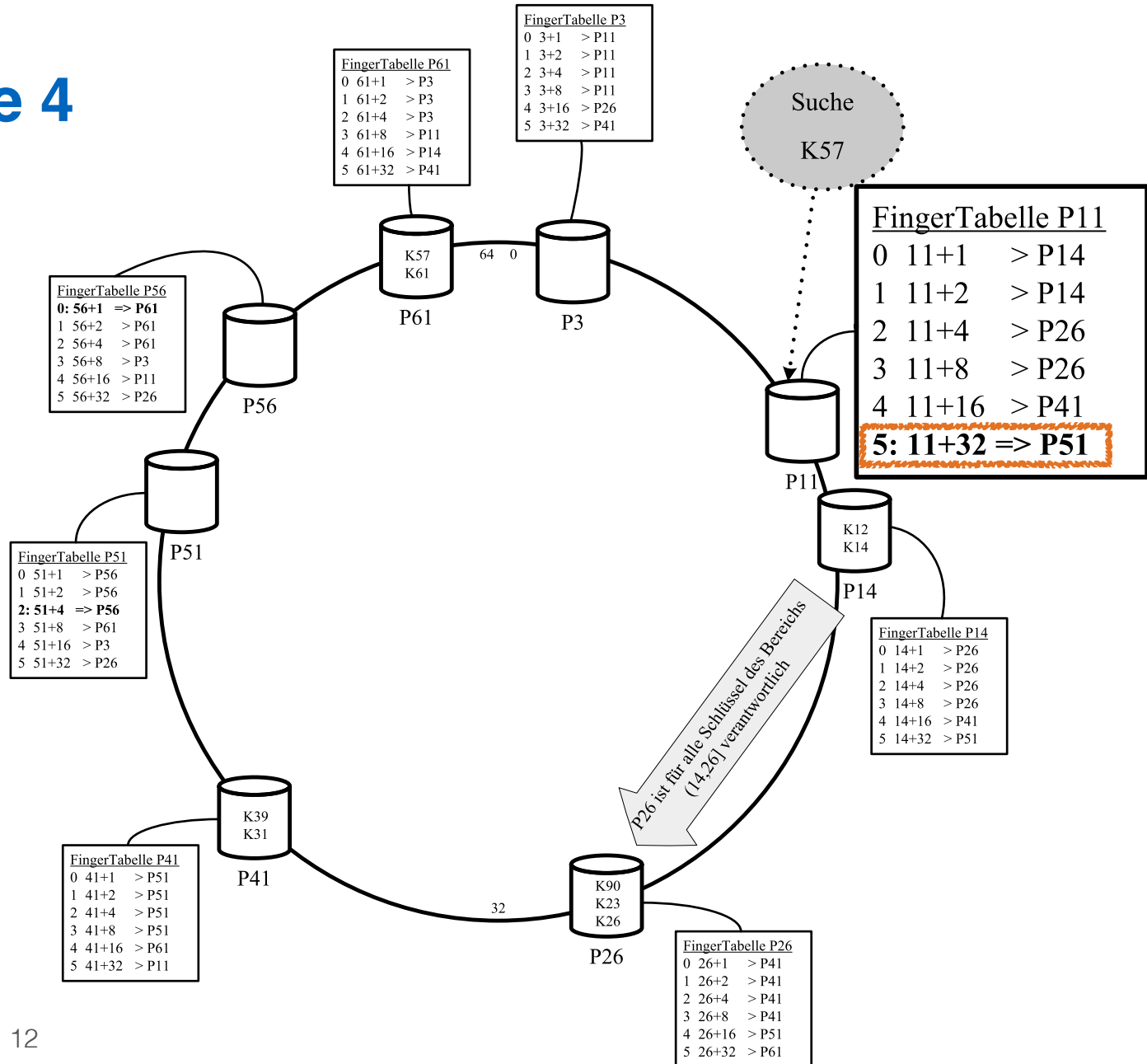
# Aufgabe 4





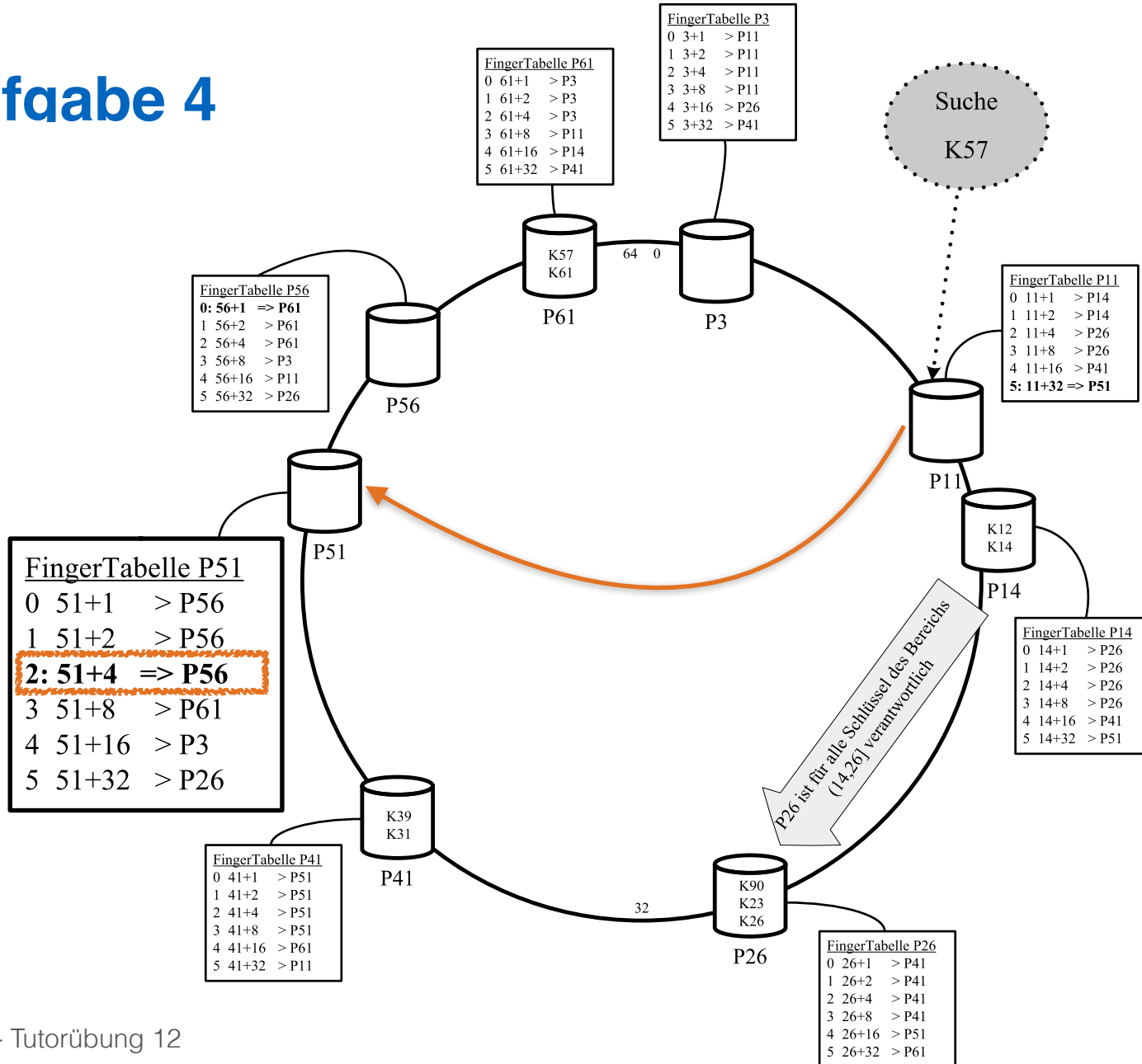


# Aufgabe 4



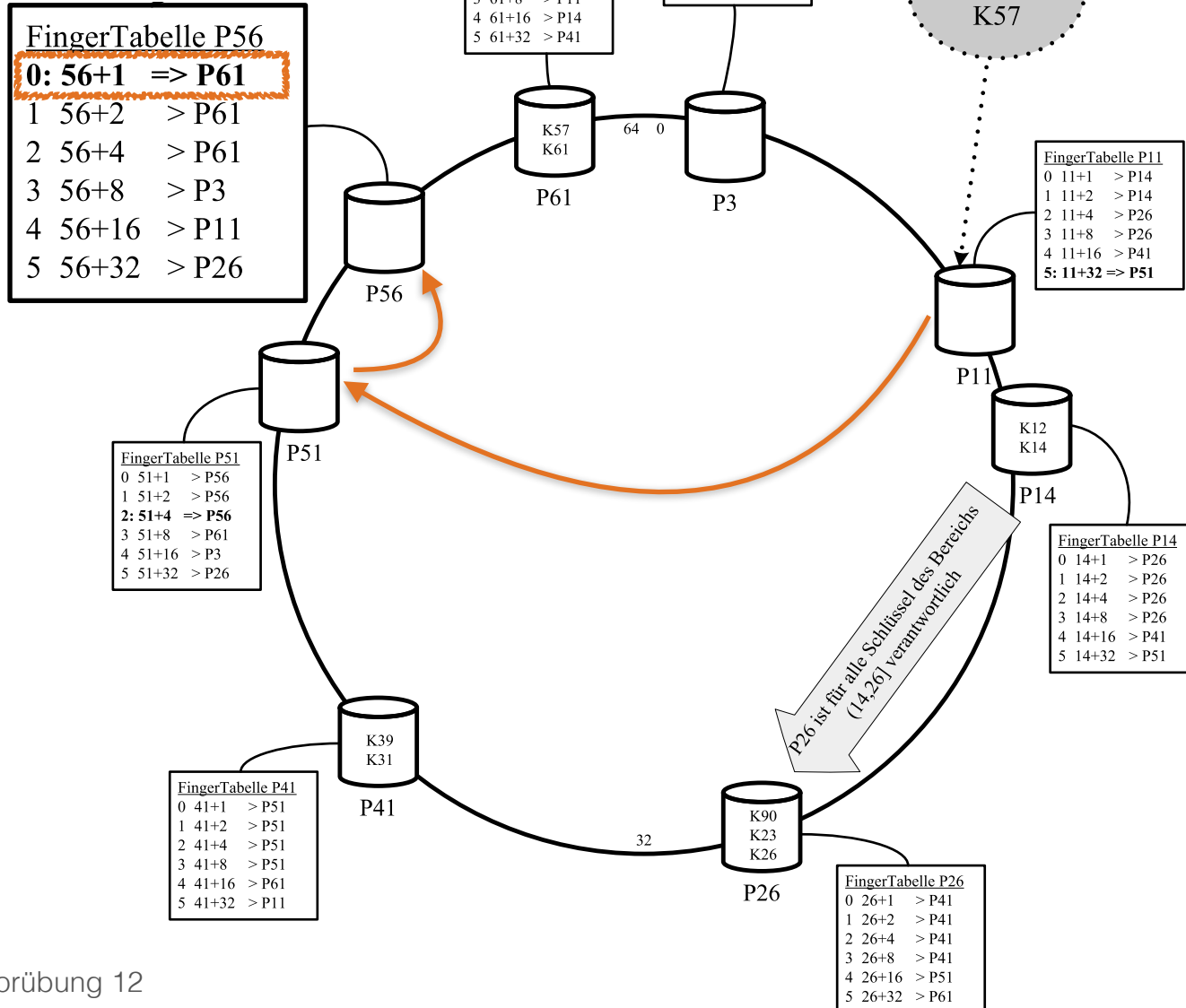


# Aufgabe 4



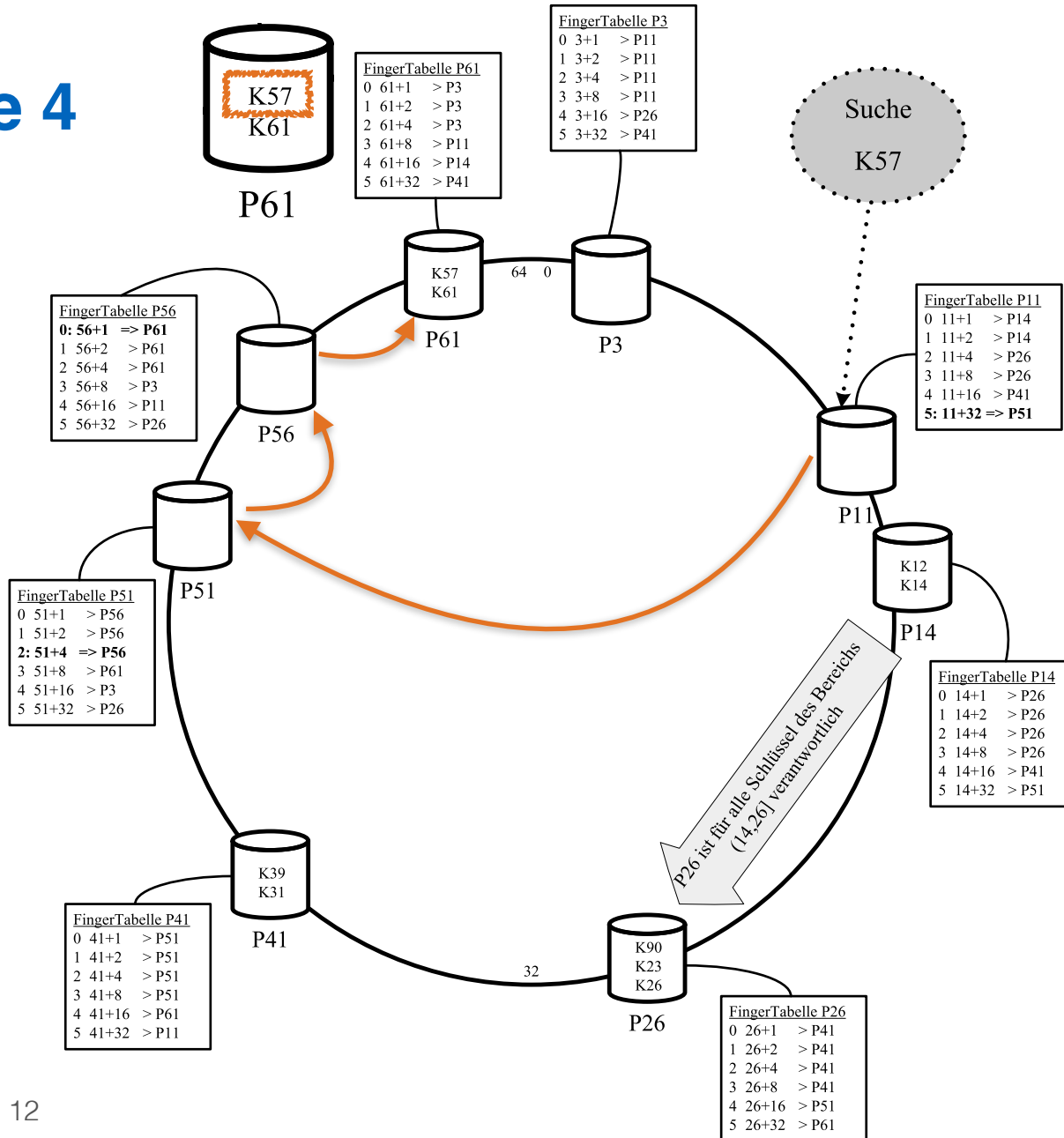


# Aufgabe 4





# Aufgabe 4



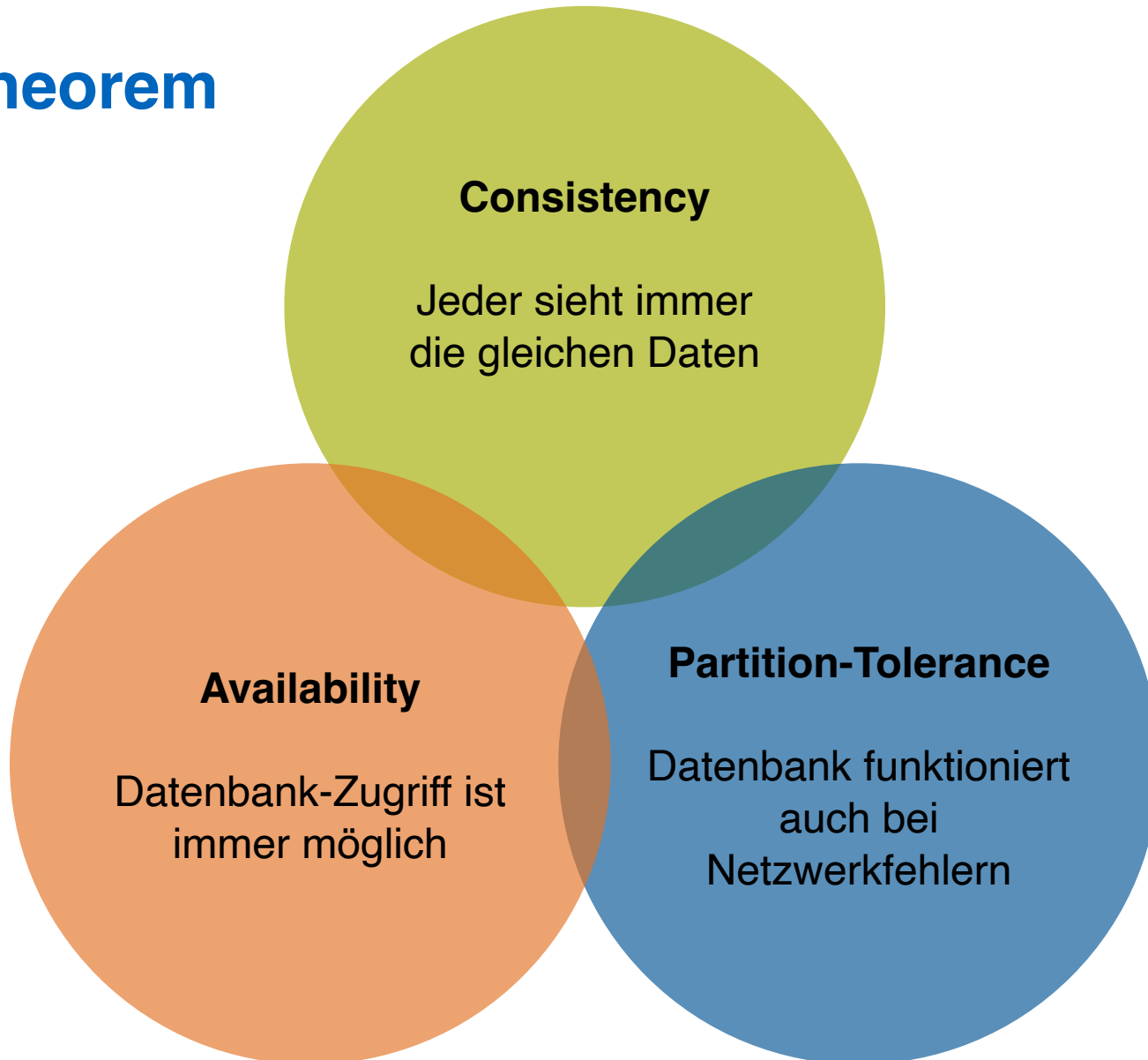


## Aufgabe 5

Skizzieren Sie die Vorgehensweise beim Hinzufügen eines neuen Peers im Chord Netzwerk. Als Beispiel nehmen Sie die Hinzunahme eines Peers P33 in dem Beispiel-Netzwerk aus Abbildung 2.



# CAP Theorem





## Aufgabe 6

Zum CAP-Theorem hieß es in der Vorlesung, dass in verteilten Systemen nur zwei der drei “Wünsche” (Konsistenz, Verfügbarkeit und Partitionstoleranz) gleichzeitig erfüllbar sind.

Welche der drei Kombinationen CA, CP, und AP sind jedoch sehr ähnlich?



**Fragen?**