

Grundlagen von Datenbanken

B-Bäume, B*-Bäume
Normalisierung



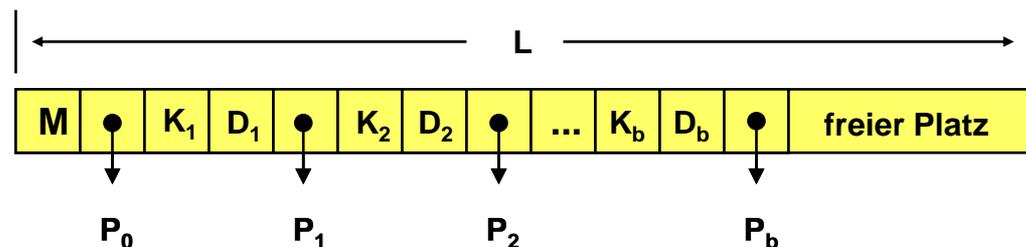
B-Bäume

Definition: Seien k, h ganze Zahlen, $h > 0, k > 0$.

Ein B-Baum B der Klasse $\tau(k, h)$ ist entweder ein leerer Baum oder ein geordneter Suchbaum mit folgenden Eigenschaften:

- Jeder Pfad von der Wurzel zu einem Blatt hat die gleiche Länge $h-1$.
- Jeder Knoten außer der Wurzel und den Blättern hat mindestens $k+1$ Söhne.
- Die Wurzel ist ein Blatt oder hat mindestens 2 Söhne.
- Jeder Knoten hat höchstens $2k+1$ Söhne.
- Jeder Knoten mit Ausnahme der Wurzel hat mindestens k und höchstens $2k$ Einträge.

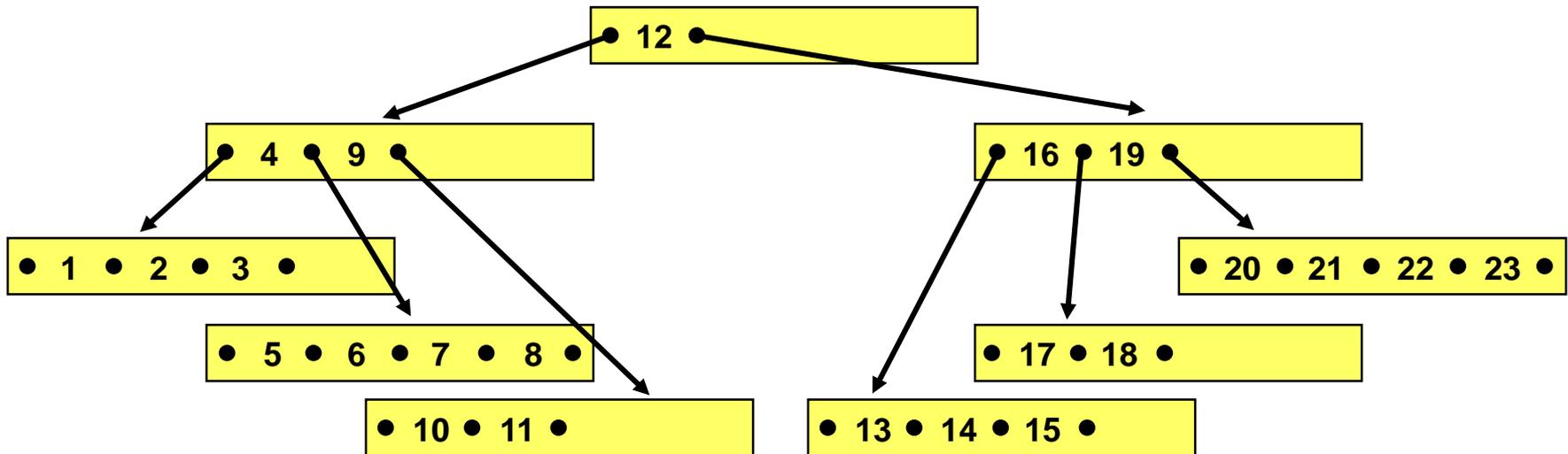
Knotenformat:



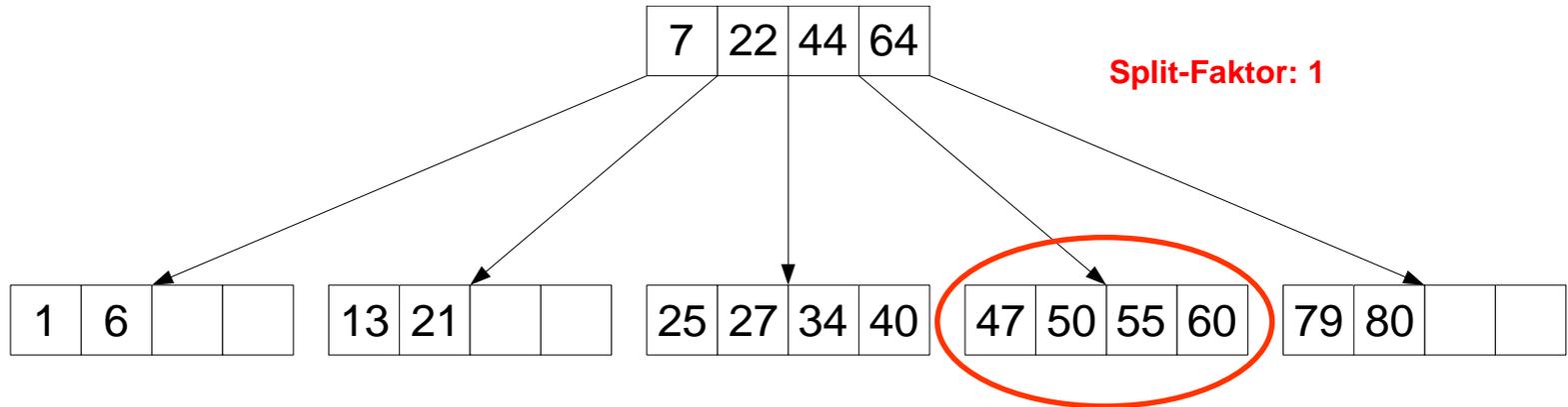
B-Bäume

- In jedem Knoten stehen die Schlüssel in aufsteigender Ordnung mit $K_1 < K_2 < \dots < K_b$.
- Jeder Schlüssel hat eine Doppelrolle als Identifikator eines Datensatzes und als Wegweiser im Baum.

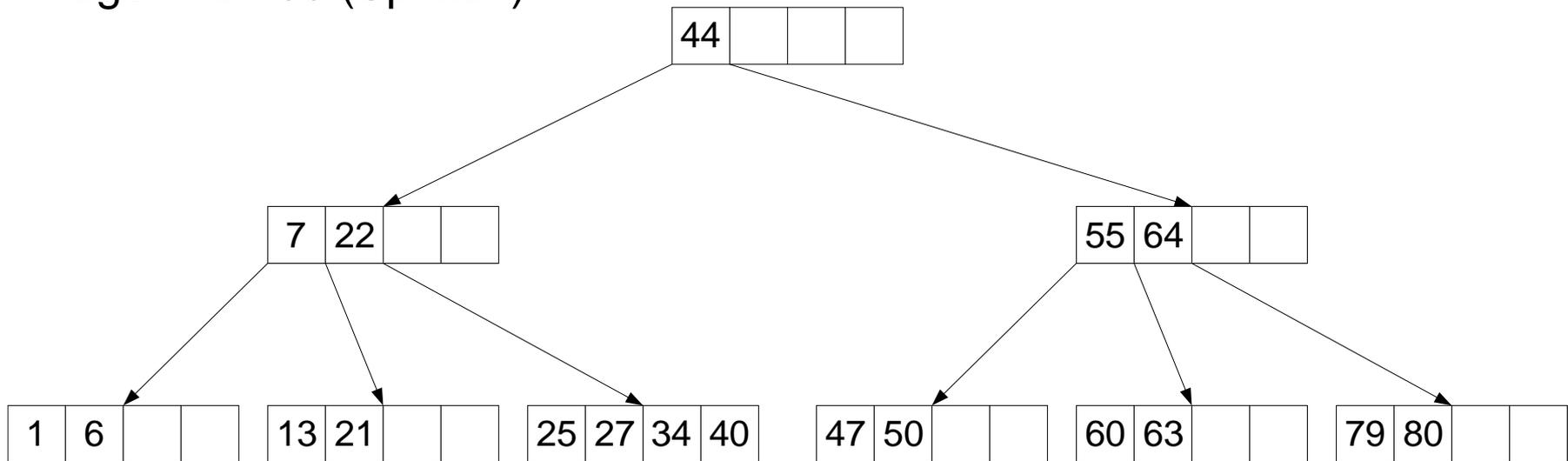
B-Baum der Klasse $t(2,3)$:



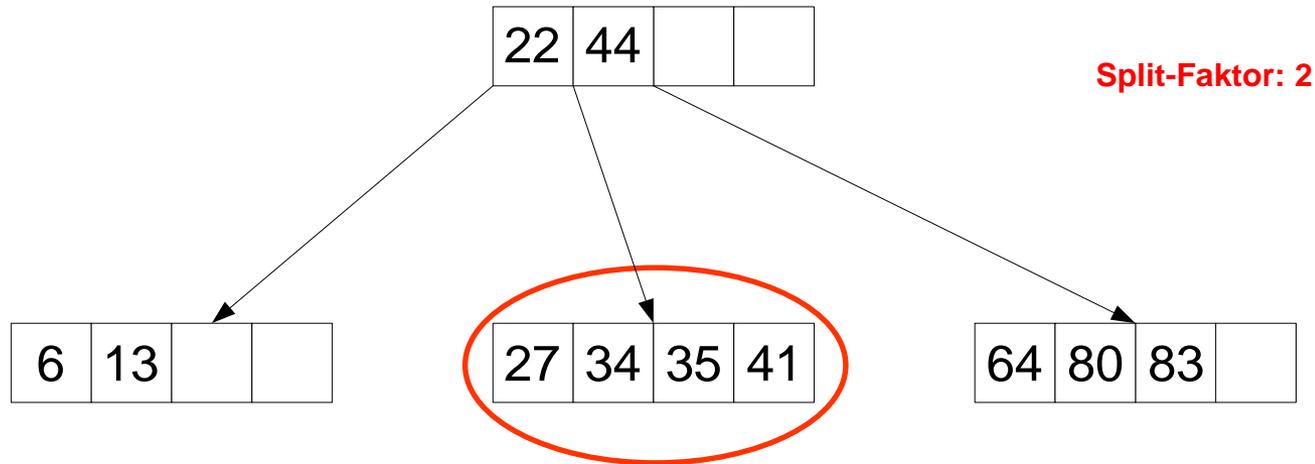
B-Bäume (Beispiel-Splitten)



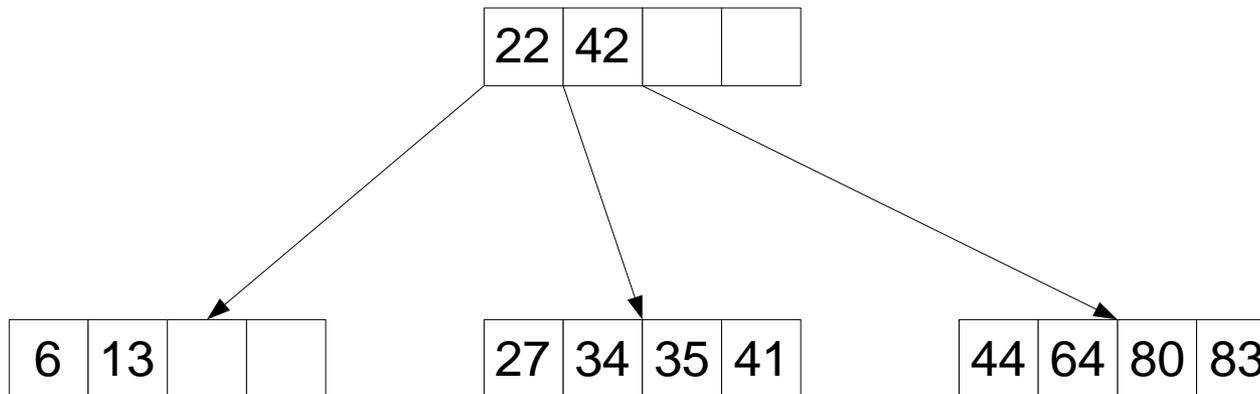
Einfügen von 63 (Splitten)



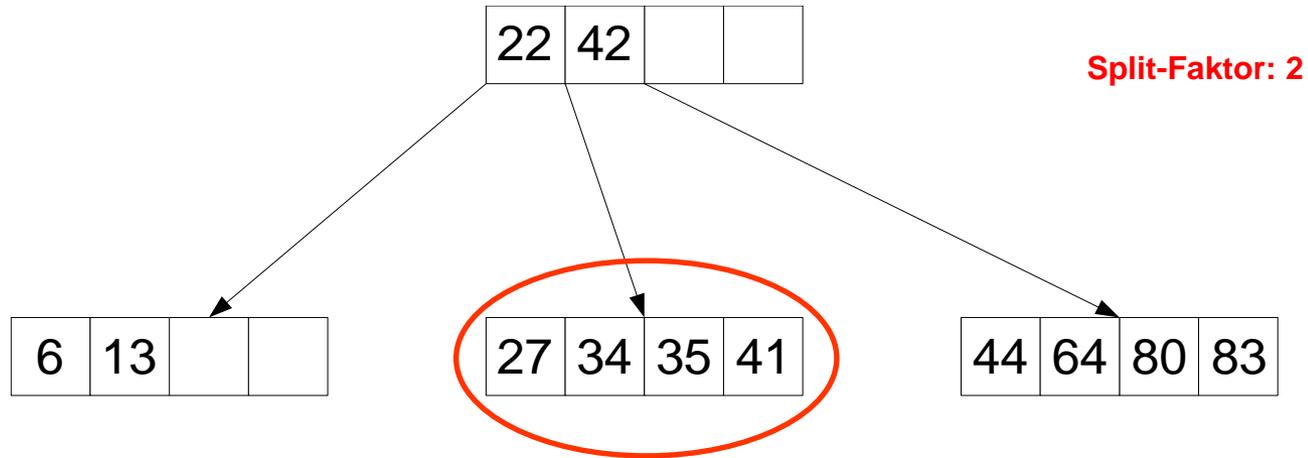
B-Bäume (Beispiel-Einfügen bei doppeltem Überlauf)



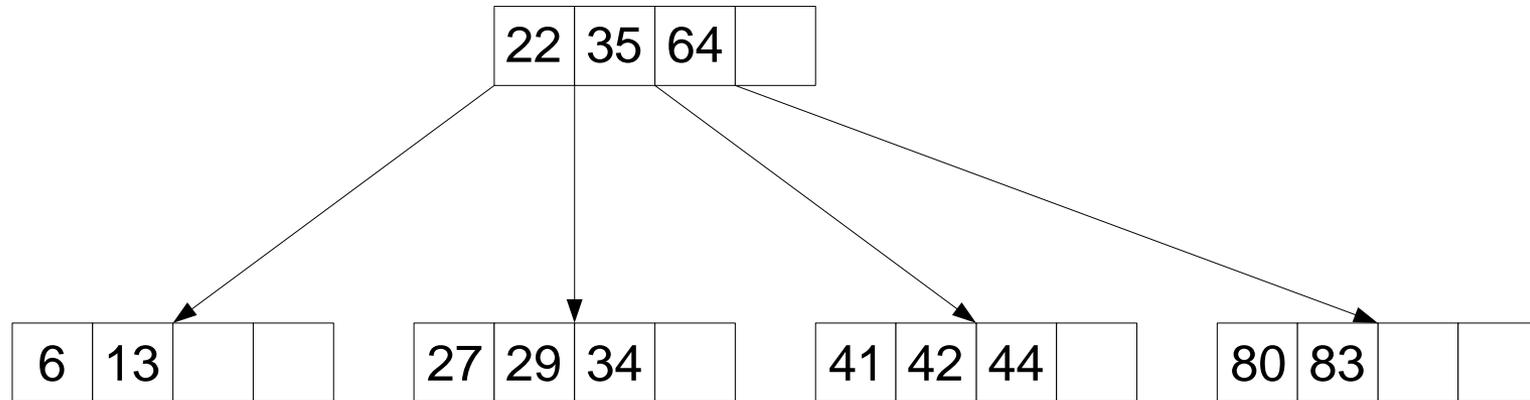
Einfügen von 42 (Überlauf)



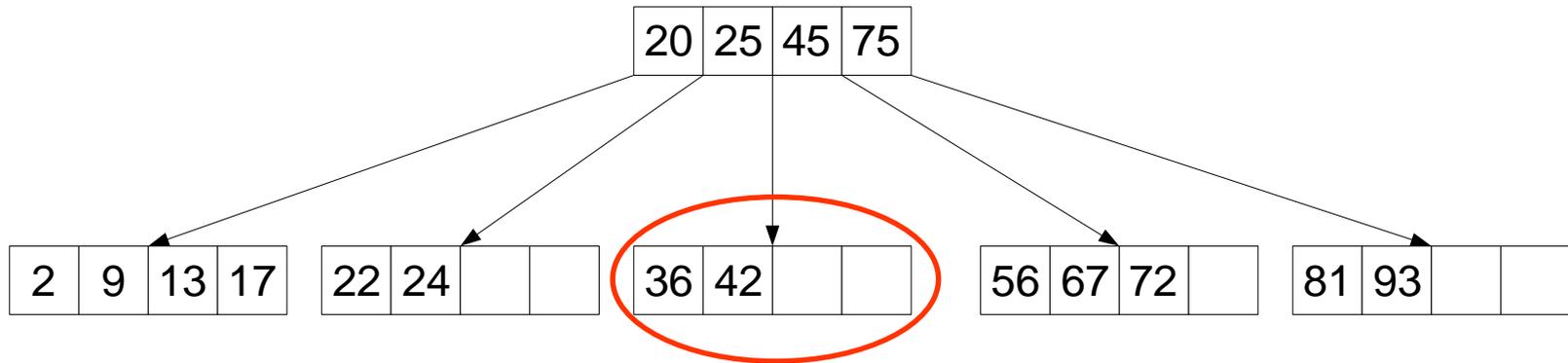
B-Bäume (Beispiel-Einfügen bei doppeltem Überlauf)



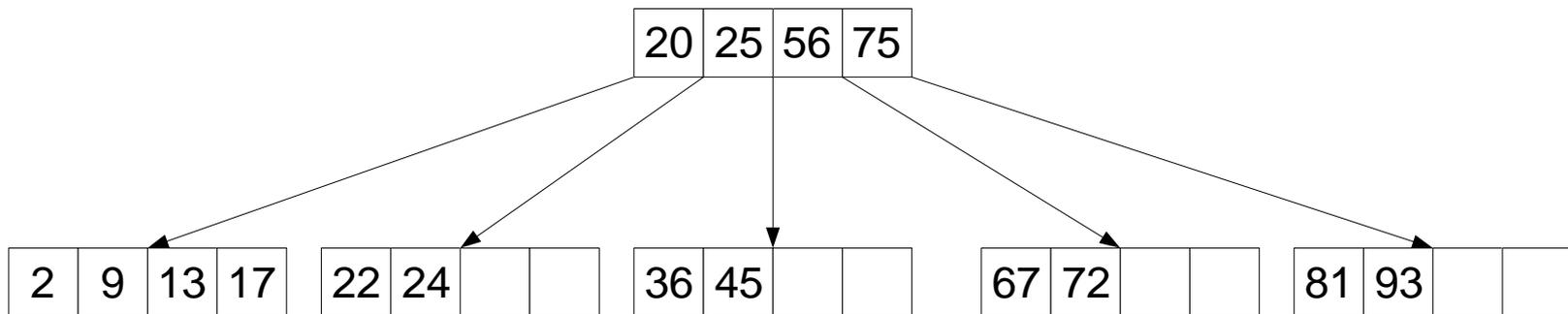
Einfügen von 29 (Splitten)



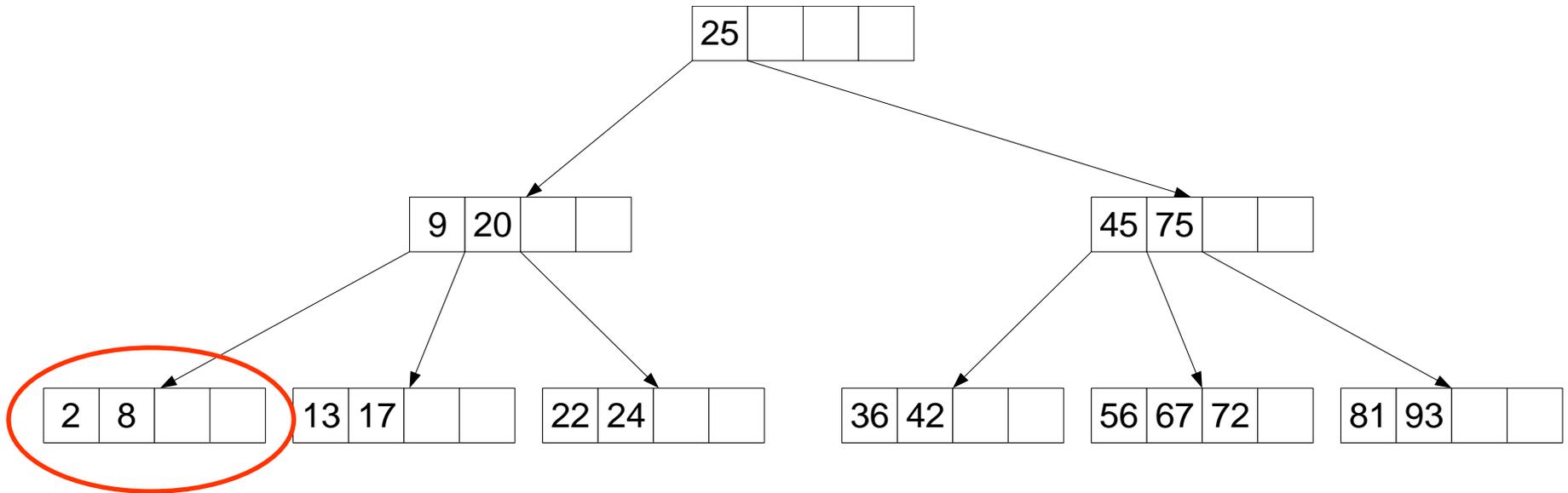
B-Bäume (Beispiel-Ausgleichen)



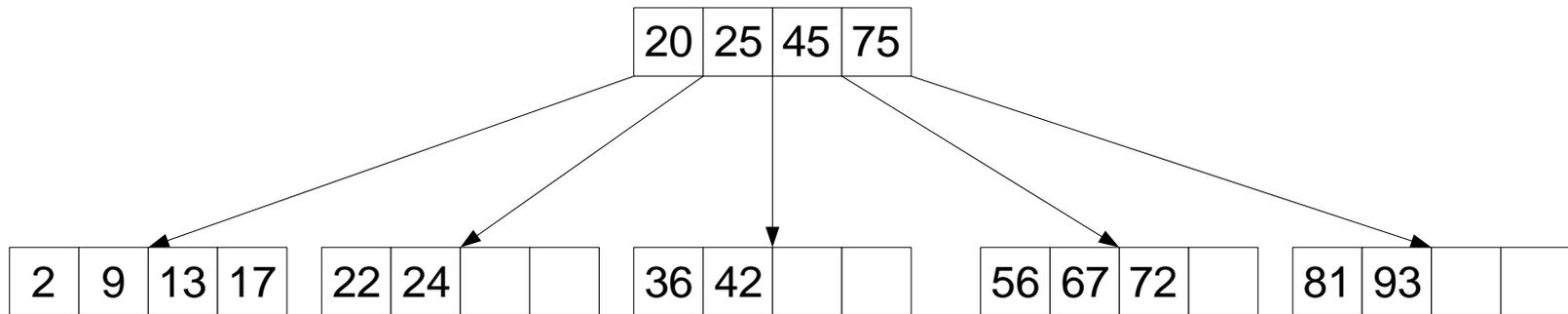
Löschen von 42 (Ausgleichen)



B-Bäume (Beispiel-Mischen)



Löschen von 8 (Mischen)



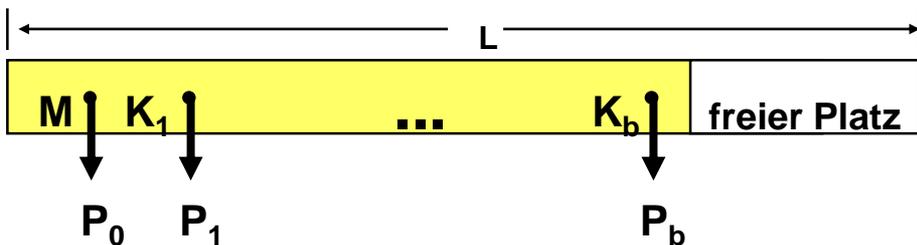
B*-Bäume

Definition: Seien k , k^* und h^* ganze Zahlen, $h^* \geq 0$, $k, k^* > 0$.

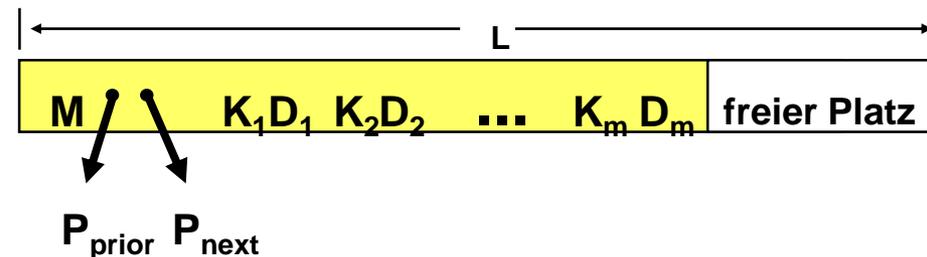
Ein B*-Baum B der Klasse $t(k, k^*, h^*)$ ist entweder ein leerer Baum oder ein geordneter Suchbaum, für den gilt:

- Jeder Pfad von der Wurzel zu einem Blatt besitzt die gleiche Länge h^*-1 .
- Jeder Knoten außer der Wurzel und den Blättern hat mindestens $k+1$ Söhne, die Wurzel mindestens 2 Söhne, außer wenn sie ein Blatt ist.
- Jeder innere Knoten hat höchstens $2k+1$ Söhne.
- Jeder Blattknoten mit Ausnahme der Wurzel als Blatt hat mindestens k^* und höchstens $2k^*$ Einträge.

Innere Knotenformat:



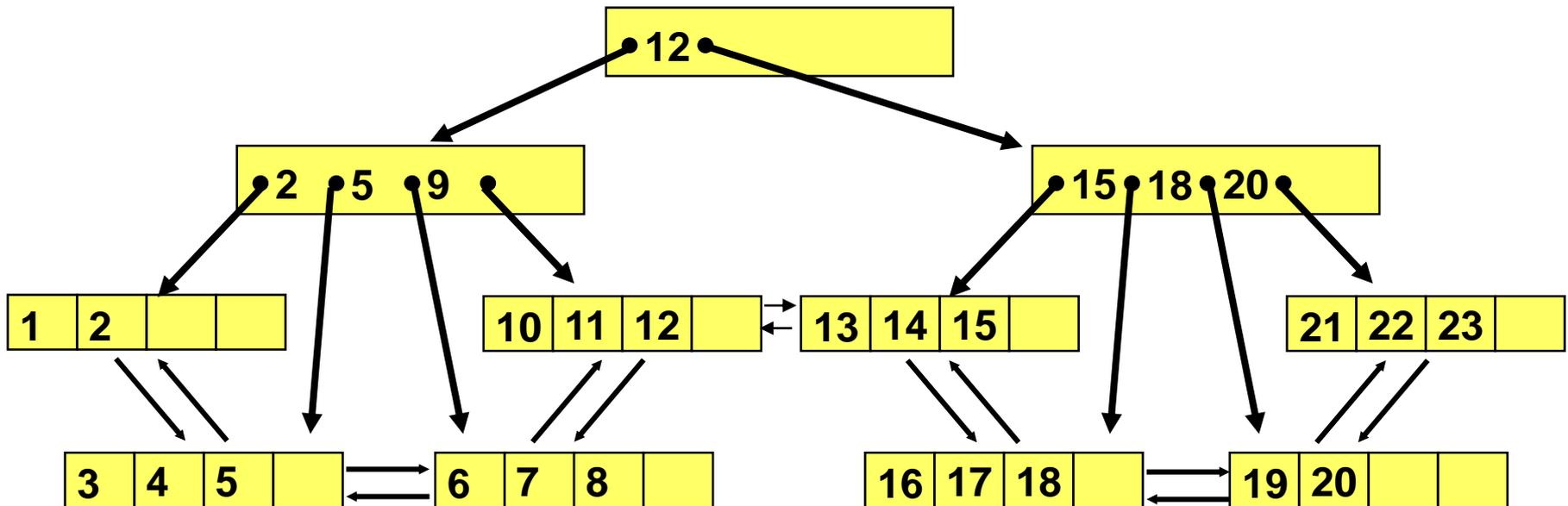
Blattknotenformat:



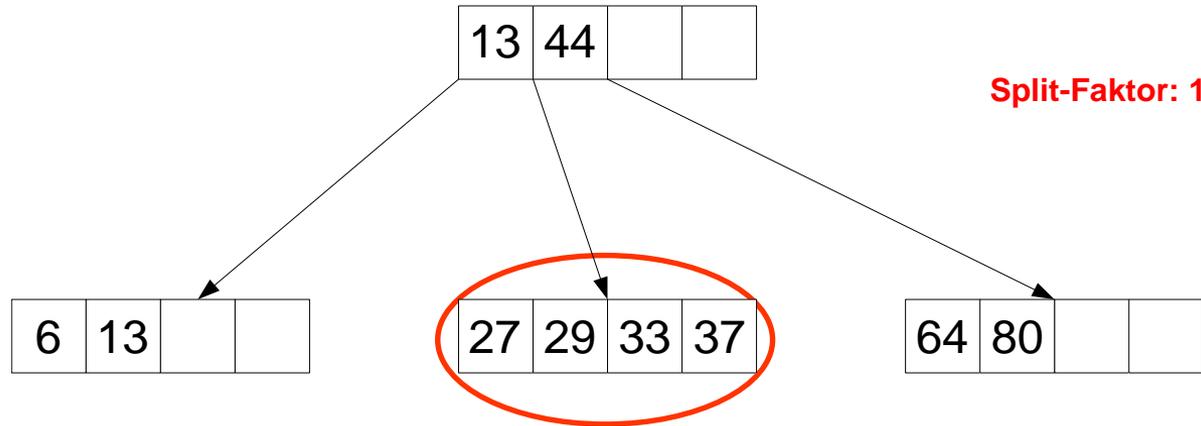
B*-Bäume (Erklärungsmodell)

Der B*-Baum lässt sich auffassen als eine gekettete sequentielle Datei von Blättern, die einen Indexteil besitzt, der selbst ein B-Baum ist. Im Indexteil werden insbes. beim Split-Vorgang die Operationen des B-Baums eingesetzt.

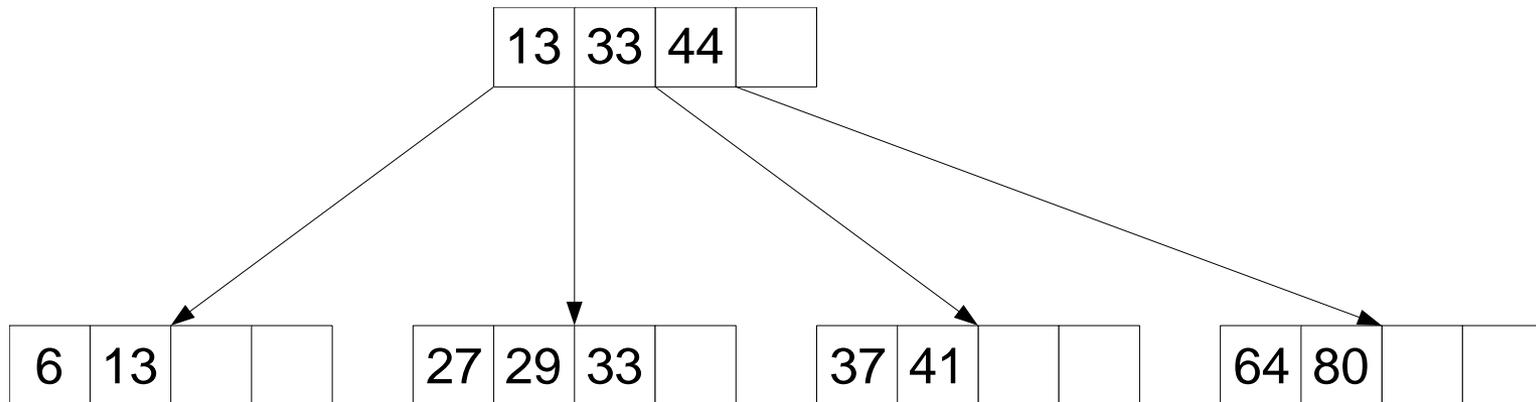
B*-Baum der Klasse $t(3, 2, 3)$:



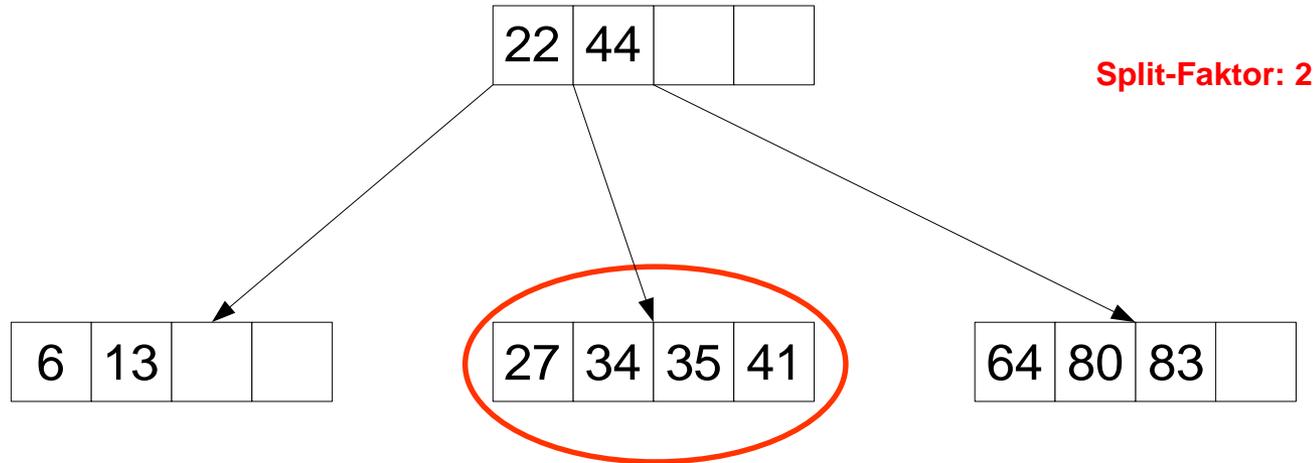
B*-Bäume (Beispiel-Splitten)



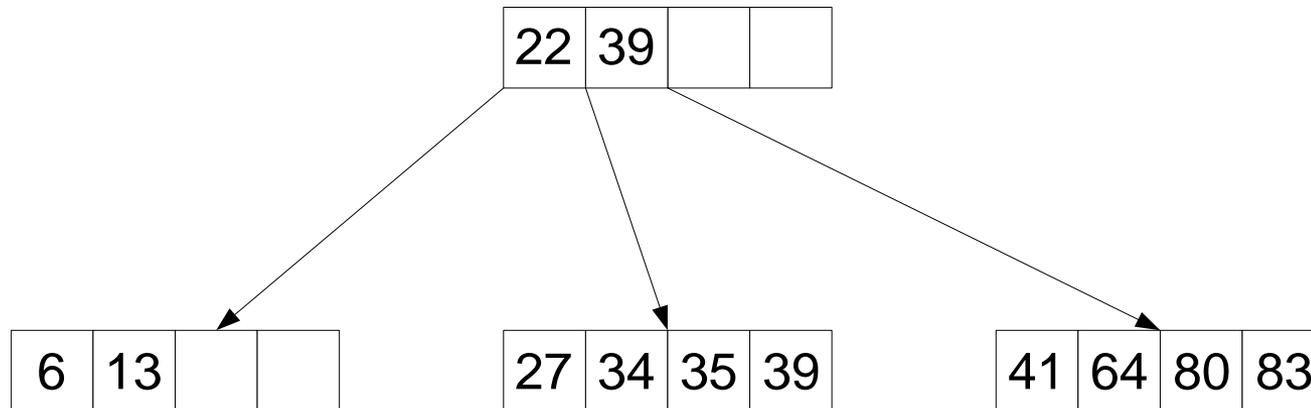
Einfügen von 41 (Splitten)



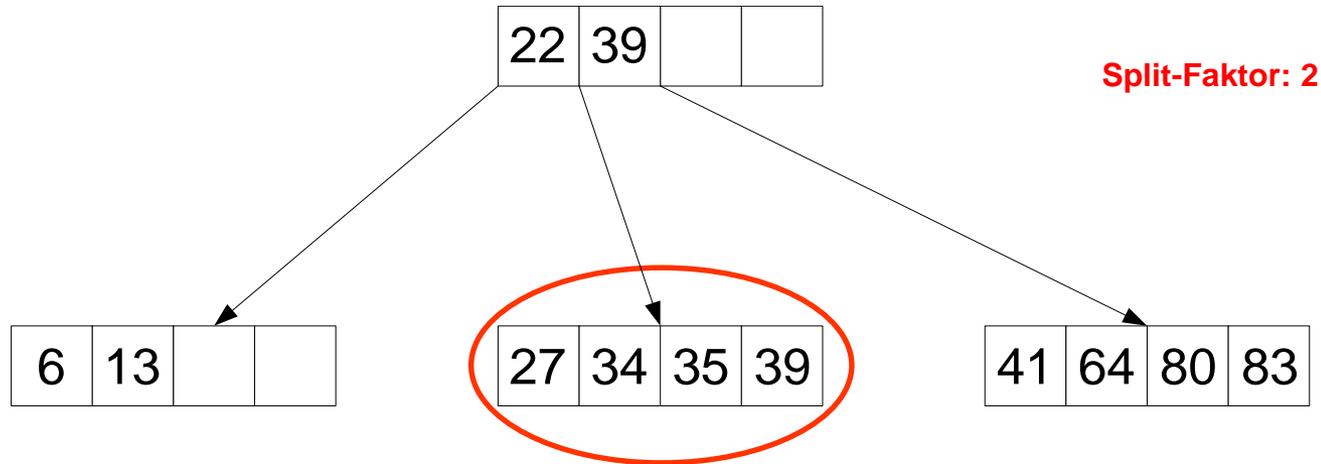
B*-Bäume (Beispiel-Einfügen bei doppeltem Überlauf)



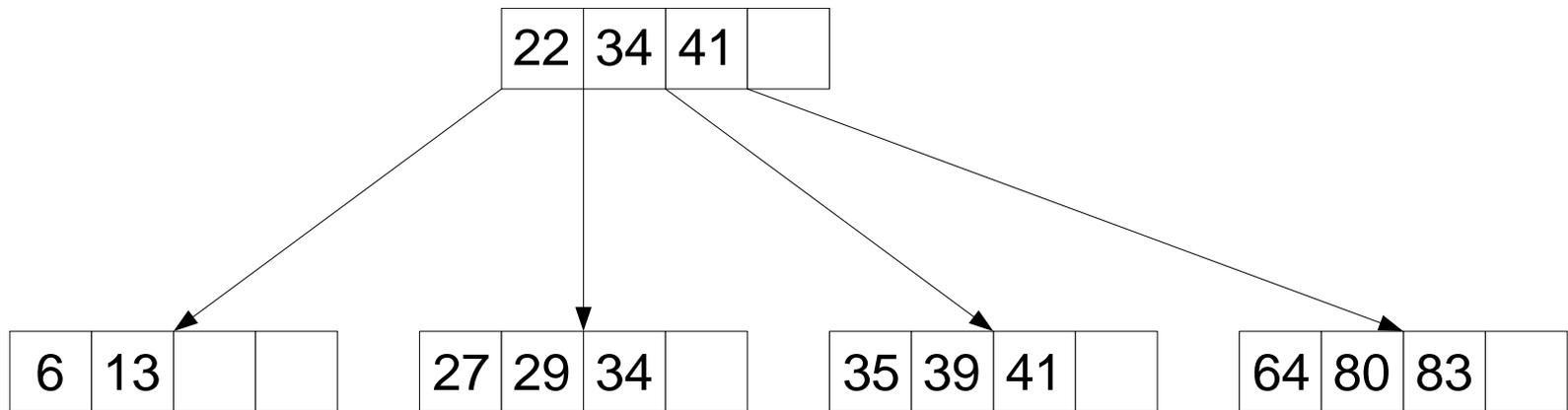
Einfügen von 39 (Überlauf)



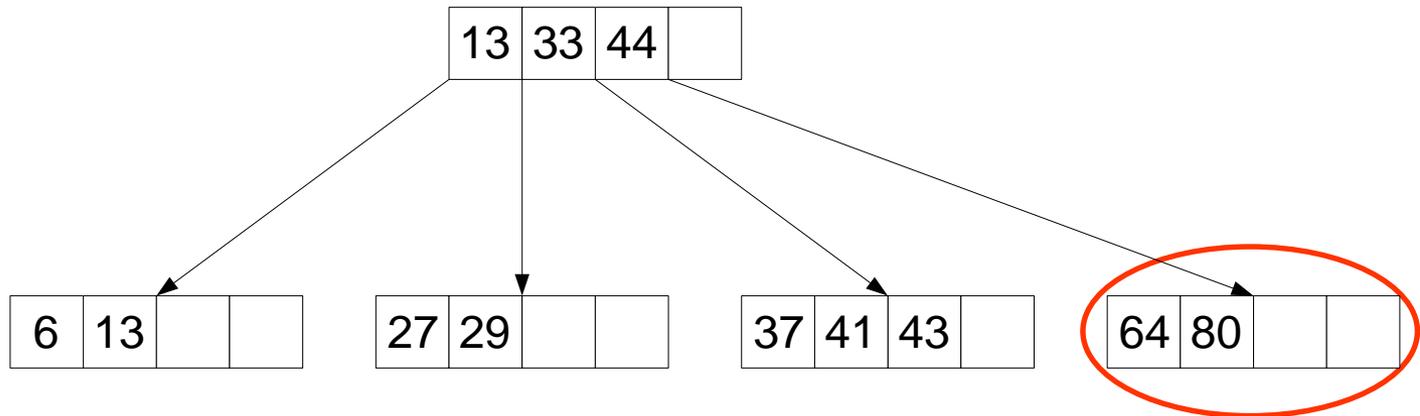
B*-Bäume (Beispiel-Einfügen bei doppeltem Überlauf)



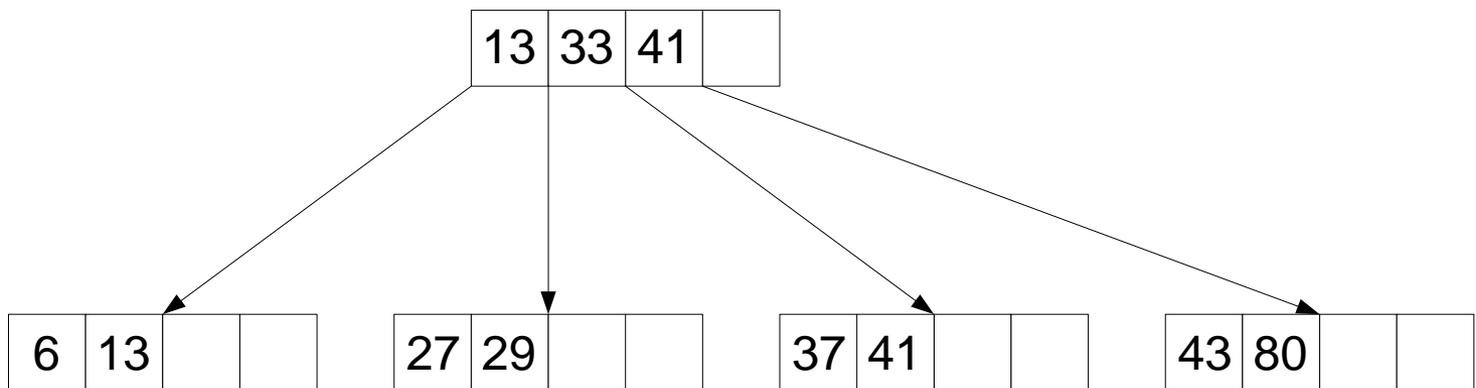
Einfügen von 29 (Splitten)



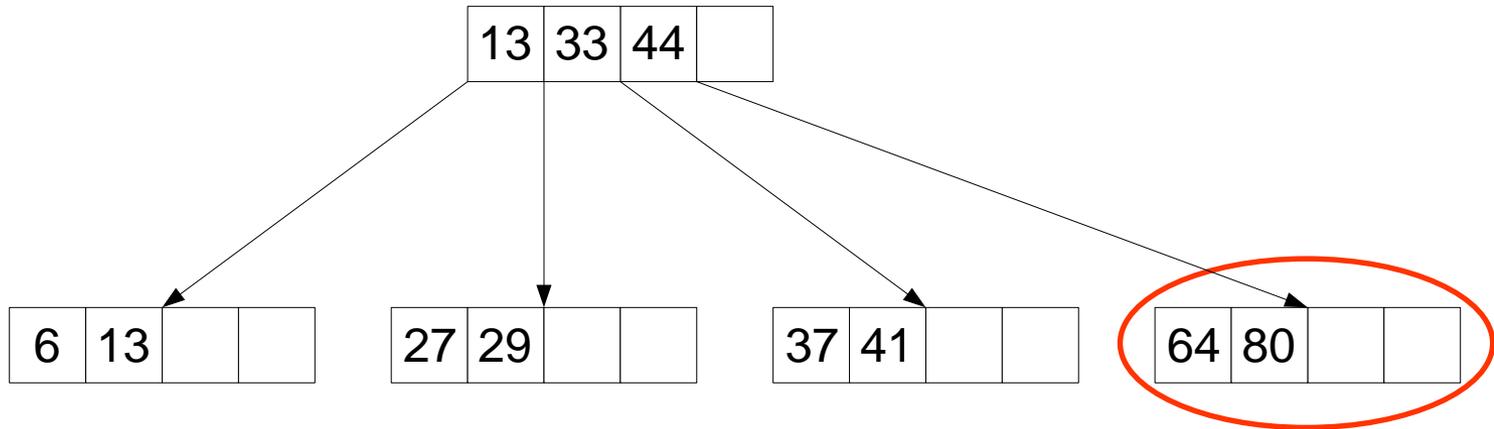
B*-Bäume (Beispiel-Ausgleichen)



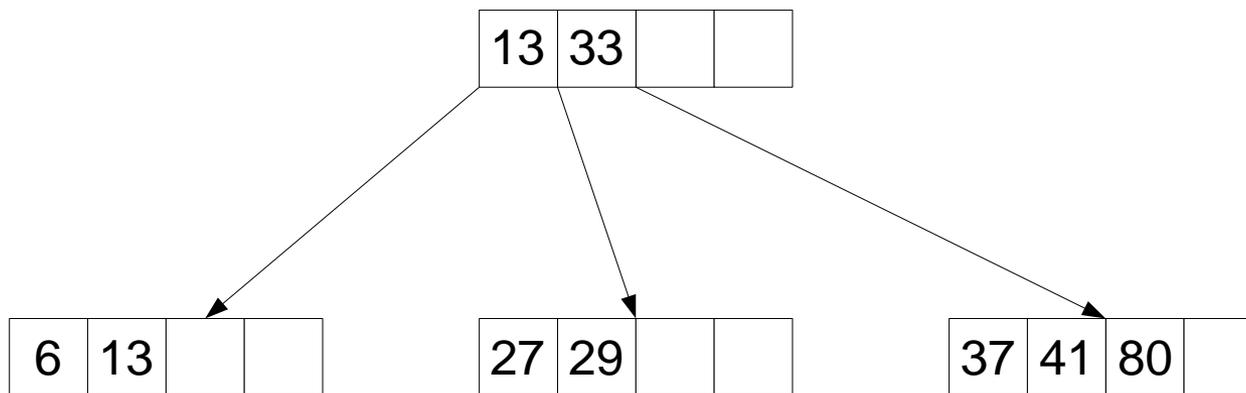
Löschen von 64 (Ausgleichen)



B*-Bäume (Beispiel-Mischen)



Löschen von 64 (Mischen)



Normalisierung

Änderungsanomalien

Bücherei (KID, Signatur, KName, ISBN, Autor, Titel, Auflage, Bestand, Ausleihdatum, Leihdauer)

KID, Signatur → Ausleihdatum, Leihdauer

KID → KName

Signatur → ISBN

ISBN → Autor, Titel

Autor, Titel → ISBN

ISBN, Auflage → Bestand

Autor, Titel, Auflage → Bestand

KID	Signatur	KNname	ISBN	Autor	Titel	Auflage	Bestand	Ausleihdatum	Leihdauer
14	1234-35	Meier	3540421335	Härder	Datenbanksysteme	1	5	29.01.07	30
2	4711-12	Müller	1558601902	Gray	Transaction Processing	1	12	01.07.06	365
2	0815-13	Müller	3540421335	Härder	Datenbanksysteme	2	10	01.07.06	365
7	9999-99	Schmidt	0130384887	Tanenbaum	Computer Networks	1	10	02.01.07	30

Normalisierung

Änderungsanomalien

Einfüge-Anomalie:

Es ist nicht möglich ein neues Buch einzufügen, wenn dieses nicht von einem Kunden ausgeliehen wird.

ODER

Es ist nicht möglich, einen Kunden anzulegen ohne dass dieser ein Buch ausleiht.

KID	Signatur	KNname	ISBN	Autor	Titel	Auflage	Bestand	Ausleihdatum	Leihdauer
14	1234-35	Meier	3540421335	Härder	Datenbanksysteme	1	5	29.01.07	30
2	4711-12	Müller	1558601902	Gray	Transaction Processing	1	12	01.07.06	365
2	0815-13	Müller	3540421335	Härder	Datenbanksysteme	2	10	01.07.06	365
7	9999-99	Schmidt	0130384887	Tanenbaum	Computer Networks	1	10	02.01.07	30

Normalisierung

Änderungsanomalien

Lösch-Anomalie:

Wenn der Kunde das letzte Buch zurückgibt, gehen auch die Kundeninformationen verloren.

ODER

Mit dem Löschen eines Kunden gehen auch die Informationen über ein Buch verloren, wenn dies von keinem anderen Kunden ausgeliehen ist.

KID	Signatur	KNname	ISBN	Autor	Titel	Auflage	Bestand	Ausleihdatum	Leihdauer
14	1234-35	Meier	3540421335	Härder	Datenbanksysteme	1	5	29.01.07	30
2	4711-12	Müller	1558601902	Gray	Transaction Processing	1	12	01.07.06	365
2	0815-13	Müller	3540421335	Härder	Datenbanksysteme	2	10	01.07.06	365
7	9999-99	Schmidt	0130384887	Tanenbaum	Computer Networks	1	10	02.01.07	30

Normalisierung

Änderungsanomalien

Modifikations-Anomalie: Um den Titel eines Buches zu korrigieren, müssen die Tupel für alle Leihverhältnisse desselben Buches aktualisiert werden.

ODER:

Ändert sich der Name eines Kunden, müssen die Tupel für alle Ausleihverhältnisse des Kunden aktualisiert werden.

KID	Signatur	KNname	ISBN	Autor	Titel	Auflage	Bestand	Ausleihdatum	Leihdauer
14	1234-35	Meier	3540421335	Härder	Datenbanksysteme	1	5	29.01.07	30
2	4711-12	Müller	1558601902	Gray	Transaction Processing	1	12	01.07.06	365
2	0815-13	Müller	3540421335	Härder	Datenbanksysteme	2	10	01.07.06	365
7	9999-99	Schmidt	0130384887	Tanenbaum	Computer Networks	1	10	02.01.07	30

Normalisierung

Normalformen

- Funktionale Abhängigkeiten:
(die Werte auf X bestimmen die Werte auf Y eindeutig)

$$X \rightarrow Y$$

$$\forall t_1, t_2 \in R: t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$$

KID, Signatur \rightarrow Ausleihdatum, Leihdauer

KID \rightarrow KName

Signatur \rightarrow ISBN

ISBN \rightarrow Autor, Titel

Autor, Titel \rightarrow ISBN

ISBN, Auflage \rightarrow Bestand

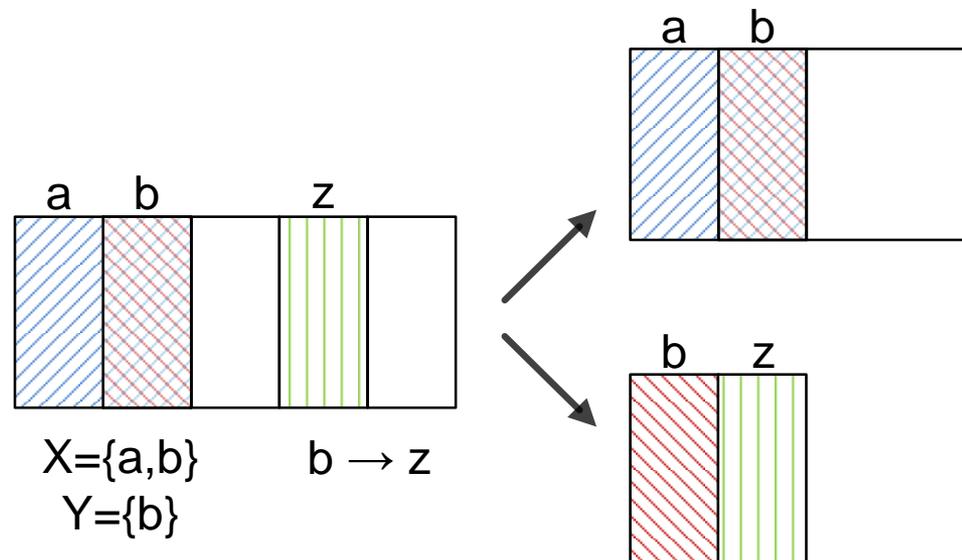
Autor, Titel, Auflage \rightarrow Bestand

Normalisierung

Normalformen

1. NF alle Attribute sind atomar.
2. NF kein Nicht-Primärattribut ist partiell von einem Schlüsselkandidaten abhängig.

NPA z, Schlüsselkandidat X $\nexists Y : (Y \subset X \wedge Y \rightarrow z)$



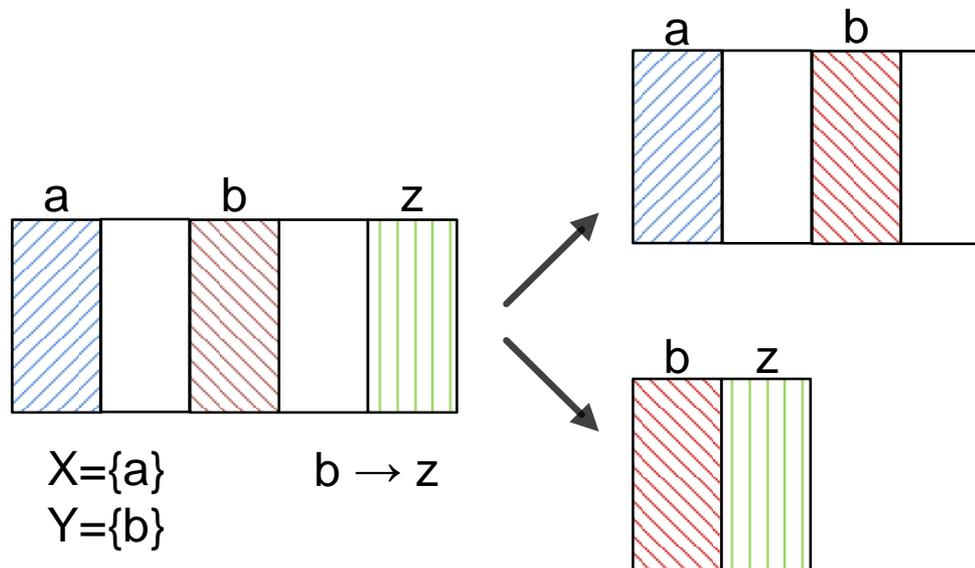
Normalisierung

Normalformen

3. NF kein Nicht-Primärattribut ist transitiv von einem Schlüsselkandidaten abhängig.

NPA z, Schlüsselkandidat X

$$\exists Y : (X \rightarrow Y \wedge \neg(Y \rightarrow X) \wedge Y \rightarrow z \wedge z \notin Y)$$



Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

a) Fkt. Abhängigkeiten

FA1: Name → Straße

FA2: Name, Datum → Treffpunkt

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

a) Fkt. Abhängigkeiten

FA1: Name \rightarrow Straße

FA2: Name, Datum \rightarrow Treffpunkt

Schlüsselkandidaten:

Name, Datum

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

a) Fkt. Abhängigkeiten

FA1: Name → Straße

FA2: Name, Datum → Treffpunkt

Schlüsselkandidaten:

Name, Datum

Nicht-Primärattribute:

Straße, Treffpunkt

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

a) Fkt. Abhängigkeiten

FA1: Name → Straße

FA2: Name, Datum → Treffpunkt

Schlüsselkandidaten:

Name, Datum

Nicht-Primärattribute:

Straße, Treffpunkt

Normalformen:

1NF, aber da **Straße** partiell von **Name, Datum** abhängt ist die Relation bzgl. dieser Menge an funktionellen Abhängigkeiten nicht in 2NF. Folglich auch nicht in 3NF.

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

b) Fkt. Abhängigkeiten

FA1: Name, Straße, Datum → Treffpunkt

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

b) Fkt. Abhängigkeiten

FA1: Name, Straße, Datum → Treffpunkt

Schlüsselkandidaten:

Name, Straße, Datum

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

b) Fkt. Abhängigkeiten

FA1: Name, Straße, Datum → Treffpunkt

Schlüsselkandidaten:

Name, Straße, Datum

Nicht-Primärattribute:

Treffpunkt

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

b) Fkt. Abhängigkeiten

FA1: Name, Straße, Datum → Treffpunkt

Schlüsselkandidaten:

Name, Straße, Datum

Nicht-Primärattribute:

Treffpunkt

Normalformen:

Da nur eine funktionale Abhängigkeit existiert, kann es weder partielle noch transitive Abhängigkeiten geben. Die Relation ist bzgl. dieser Menge an funktionellen Abhängigkeiten somit in 3NF.

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

c) Fkt. Abhängigkeiten

FA1: Name, Straße → Datum, Treffpunkt

FA2: Datum, Treffpunkt → Name, Straße

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

c) Fkt. Abhängigkeiten

FA1: Name, Straße → Datum, Treffpunkt

FA2: Datum, Treffpunkt → Name, Straße

Schlüsselkandidaten:

Name, Straße oder Datum, Treffpunkt

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

c) Fkt. Abhängigkeiten

FA1: Name, Straße → Datum, Treffpunkt

FA2: Datum, Treffpunkt → Name, Straße

Schlüsselkandidaten:

Name, Straße oder Datum, Treffpunkt

Nicht-Primärattribute:

∅

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

c) Fkt. Abhängigkeiten

FA1: Name, Straße \rightarrow Datum, Treffpunkt

FA2: Datum, Treffpunkt \rightarrow Name, Straße

Schlüsselkandidaten:

Name, Straße oder Datum, Treffpunkt

Nicht-Primärattribute:

∅

Normalformen:

Da es keine Nicht-Primärattribute gibt, muss die Relation bzgl. dieser Menge an funktionellen Abhängigkeiten in 3NF sein.

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

d) Fkt. Abhängigkeiten

FA1: Name, Straße → Datum, Treffpunkt

FA2: Datum → Treffpunkt

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

d) Fkt. Abhängigkeiten

FA1: Name, Straße → Datum, Treffpunkt

FA2: Datum → Treffpunkt

Schlüsselkandidaten:

Name, Straße

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

d) Fkt. Abhängigkeiten

FA1: Name, Straße → Datum, Treffpunkt

FA2: Datum → Treffpunkt

Schlüsselkandidaten:

Name, Straße

Nicht-Primärattribute:

Datum, Treffpunkt

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

d) Fkt. Abhängigkeiten

FA1: Name, Straße → Datum, Treffpunkt

FA2: Datum → Treffpunkt

Schlüsselkandidaten:

Name, Straße

Nicht-Primärattribute:

Datum, Treffpunkt

Normalformen:

Es treten keine partiellen Abhängigkeiten auf. Die Relation ist somit in 2NF. Da **Treffpunkt** aber über **Datum** transitiv von **Name, Straße** abhängt, ist die Relation bzgl. dieser Menge an funktionellen Abhängigkeiten nicht in 3NF.

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

e) Fkt. Abhängigkeiten

FA1: Name → Datum

FA2: Datum, Treffpunkt → Straße

FA3: Straße → Name

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

e) Fkt. Abhängigkeiten

FA1: Name \rightarrow Datum

FA2: Datum, Treffpunkt \rightarrow Straße

FA3: Straße \rightarrow Name

Schlüsselkandidaten:

Datum, Treffpunkt

oder Name, Treffpunkt

oder Straße, Treffpunkt

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

e) Fkt. Abhängigkeiten

FA1: Name → Datum

FA2: Datum, Treffpunkt → Straße

FA3: Straße → Name

Schlüsselkandidaten:

Datum, Treffpunkt

oder Name, Treffpunkt

oder Straße, Treffpunkt

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

e) Fkt. Abhängigkeiten

FA1: Name → Datum

FA2: Datum, Treffpunkt → Straße

FA3: Straße → Name

Schlüsselkandidaten:

Datum, Treffpunkt

oder Name, Treffpunkt

oder Straße, Treffpunkt

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

e) Fkt. Abhängigkeiten

FA1: Name \rightarrow Datum

FA2: Datum, Treffpunkt \rightarrow Straße

FA3: Straße \rightarrow Name

Schlüsselkandidaten:

Datum, Treffpunkt
oder Name, Treffpunkt
oder Straße, Treffpunkt

Nicht-Primärattribute:

∅

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

e) Fkt. Abhängigkeiten

FA1: Name \rightarrow Datum

FA2: Datum, Treffpunkt \rightarrow Straße

FA3: Straße \rightarrow Name

Schlüsselkandidaten:

Datum, Treffpunkt
oder Name, Treffpunkt
oder Straße, Treffpunkt

Nicht-Primärattribute:

∅

Normalformen:

Da es keine Nicht-Primärattribute gibt, muss die Relation bzgl. dieser Menge an funktionellen Abhängigkeiten in 3NF sein.

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

f) Fkt. Abhängigkeiten

FA1: Name → Straße

FA2: Datum → Treffpunkt

FA3: Straße, Datum → Name

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

f) Fkt. Abhängigkeiten

FA1: Name → Straße

FA2: Datum → Treffpunkt

FA3: Straße, Datum → Name

Schlüsselkandidaten:

Name, Datum oder Straße, Datum

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

f) Fkt. Abhängigkeiten

FA1: Name → Straße

FA2: Datum → Treffpunkt

FA3: Straße, Datum → Name

Schlüsselkandidaten:

Name, Datum oder Straße, Datum

Nicht-Primärattribute:

Treffpunkt

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

f) Fkt. Abhängigkeiten

FA1: Name \rightarrow Straße

FA2: Datum \rightarrow Treffpunkt

FA3: Straße, Datum \rightarrow Name

Schlüsselkandidaten:

Name, Datum oder Straße, Datum

Nicht-Primärattribute:

Treffpunkt

Normalformen:

1NF, aber da **Treffpunkt** partiell von **Straße, Datum** und von **Name, Datum** abhängt ist die Relation bzgl. dieser Menge an funktionellen Abhängigkeiten nicht in 2NF. Folglich auch nicht in 3NF.

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

g) Fkt. Abhängigkeiten

FA1: Name, Straße → Treffpunkt

FA2: Datum → Treffpunkt

FA3: Treffpunkt → Datum

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

g) Fkt. Abhängigkeiten

FA1: Name, Straße → Treffpunkt

FA2: Datum → Treffpunkt

FA3: Treffpunkt → Datum

Schlüsselkandidaten:

Name, Straße

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

g) Fkt. Abhängigkeiten

FA1: Name, Straße → Treffpunkt

FA2: Datum → Treffpunkt

FA3: Treffpunkt → Datum

Schlüsselkandidaten:

Name, Straße

Nicht-Primärattribute:

Datum, Treffpunkt

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

g) Fkt. Abhängigkeiten

FA1: Name, Straße → Treffpunkt

FA2: Datum → Treffpunkt

FA3: Treffpunkt → Datum

Schlüsselkandidaten:

Name, Straße

Nicht-Primärattribute:

Datum, Treffpunkt

Normalformen:

Es treten keine partiellen Abhängigkeiten auf. Die Relation ist bzgl. dieser funktionellen Abhängigkeiten somit in 2NF. Da Datum aber über Treffpunkt transitiv von Name, Straße abhängt, ist die Relation bzgl. dieser Menge an funktionellen Abhängigkeiten nicht in 3NF.

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

h) Fkt. Abhängigkeiten

FA1: Treffpunkt → Name

FA2: Name → Datum, Straße, Treffpunkt

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

h) Fkt. Abhängigkeiten

FA1: Treffpunkt → Name

FA2: Name → Datum, Straße, Treffpunkt

Schlüsselkandidaten:

Treffpunkt oder Name

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

h) Fkt. Abhängigkeiten

FA1: Treffpunkt → Name

FA2: Name → Datum, Straße, Treffpunkt

Schlüsselkandidaten:

Treffpunkt oder Name

Nicht-Primärattribute:

Straße, Datum

Normalisierung

R1(Name, Straße, Datum, Treffpunkt)

h) Fkt. Abhängigkeiten

FA1: Treffpunkt \rightarrow Name

FA2: Name \rightarrow Datum, Straße, Treffpunkt

Schlüsselkandidaten:

Treffpunkt oder Name

Nicht-Primärattribute:

Straße, Datum

Normalformen:

Da beide Schlüsselkandidaten einattributig sind, kann es keine partiellen Abhängigkeiten geben. Die Relation ist bzgl. dieser funktionellen Abhängigkeiten somit in 2NF. Da es sich bei **Treffpunkt** und **Name** jeweils um einen Schlüsselkandidaten handelt und somit **Treffpunkt \leftrightarrow Name** gilt, existieren auch keine transitiven Abhängigkeiten. Die Relation ist bzgl. dieser Menge an funktionellen Abhängigkeiten daher sogar in 3NF.