

# Database Systems - Formulas

Lasse Schuirmann

March 21, 2015

THIS PAGE INTENTIONALLY LEFT BLANK.

## ER Diagramme

$kunst \overset{N}{-} \overset{M}{sein} Ausstellung$

Lesen wie:

Kunst KANN in M Ausstellungen sein.

Eine Ausstellung MUSS (1-) N Kunstgegenstände enthalten.

## Relationales Schema

Object: {[ Primärschlüssel: Typ, Andere Schluesel: Andere Typen ]}

Prädikat: {[ PrimärschlüsselVonObjekt1: Typ, PrimärschlüsselVonObjekt2: Typ, Eigenschaften: Typ ]}

## Schlüssel

### Superschlüssel

Ein Superschlüssel definiert implizit alle anderen Attribute der Relation.

### Schlüsselkandidat

Ein Schlüsselkandidat ist ein minimaler Superschlüssel.

### Primärschlüssel

Ein Primärschlüssel ist der ausgewählte Schlüsselkandidat.

## Komische Symbole

### Attributrelationen

$\alpha \rightarrow \beta \Leftrightarrow \alpha$  bestimmt eindeutig/ist Superschlüssel für  $\beta$

$\alpha \twoheadrightarrow \beta \Leftrightarrow \alpha$  ist Schlüsselkandidat für  $\beta$

## Queries

$\sigma_c(R)$  ist eine Anfrage auf die Datenbank  $R$  mit der Bedingung  $c$ . (SELECT ... FROM  $R$  WHERE  $c$ )

$\pi_L(R)$  Projektion (SELECT something AS else FROM ...)

$\cup$  Vereinigung

$\cap$  Durchschnitt

$-$  Differenz (SELECT \* FROM  $T_1$  WHERE  $ID$  NOT IN (SELECT  $ID$  FROM  $T_2$ ))

$R \times L$  Kartesisches Produkt (SELECT ... FROM  $R, L$ )

$\bowtie$  Verbund/Join (JOIN ... ON)

$\gamma$  Gruppierung (GROUP BY)

$\delta$  Duplikationselimination (SELECT DISTINCT)

$\tau$  Sortieren (ORDER BY ... [ASC/DESC])

## Volcano-Iterator-Modell

Wie python Iteratoren.

Iteratoren haben dann: open(), next(), close()

Dann:

1. open() fuer alle operatoren durchfuehren
2. next() fuer alle operatoren durchfuehren, ergebnisse durchreichen
3. letzten Schritt wiederholen bis next() eof/null zurueckgibt
4. close() fuer alle operatoren durchfuehren

## Normalformen

### 1. NF

Attribute sind atomar.

### 2. NF

Attribute sind nicht abhängig von einer echten Teilmenge eines Schlüsselkandidaten.

### 3. NF

Attribute sind ausschliesslich abhängig von dem Primärschlüssel.

## **RAID**

### **RAID 0**

Reissverschlussverfahren um Zugriffszeiten zu optimieren.

### **RAID 1**

Spiegelung - komplettbackup. Kein Performancegewinn.

### **RAID 5**

Eine Paritätsplatte, kompensiert Ausfall einer Platte. Performancegewinn durch aufteilung.

## **B+ Baum**

Ähnlich eines binären Baums, kann zur Suche benutzt werden. Ein Baum mit der Tiefe  $d$  hat an jedem Knoten  $d$  bis  $2 * d$  Unterknoten. An dem Wurzelknoten können weniger Unterknoten vorhanden sein.

Ein geclonierter B+ Baum ist im Speicher ebenso sortiert nach beliebigen Kriterien.

An den Blättern sind üblicherweise Zeiger zu dem Zieldatensatz.

## **Mergesort**

### **2-Way-Mergesort**

### **2-Phase-Multiway-Mergesort**

#### **Vorraussetzungen**

Größe der zu sortierenden Relation =  $B$  Blöcke.

Anzahl der verfügbaren Seiten im Hauptspeicher =  $M$ .

$$B \leq M^2$$

#### **Phase 1**

Relation wird blockweise in die  $M$  Seiten im Hauptspeicher gelesen und dort sortiert.

Das Ergebnis wird anhand von sortierten Läufen der Länge  $N$  in den Sekundärspeicher geschrieben.

## **Phase 2**

Sortierte Läufe werden gemischt.

Jeder Lauf wird sukzessive in eine der Seiten gelesen und in die Ausgabeseite durch Mischen sortiert ausgegeben.

Ist die Ausgabeseite voll wird sie in den Sekundärspeicher geschrieben.