12. Anfrageoptimierung

Vorlesung "Informationssysteme"

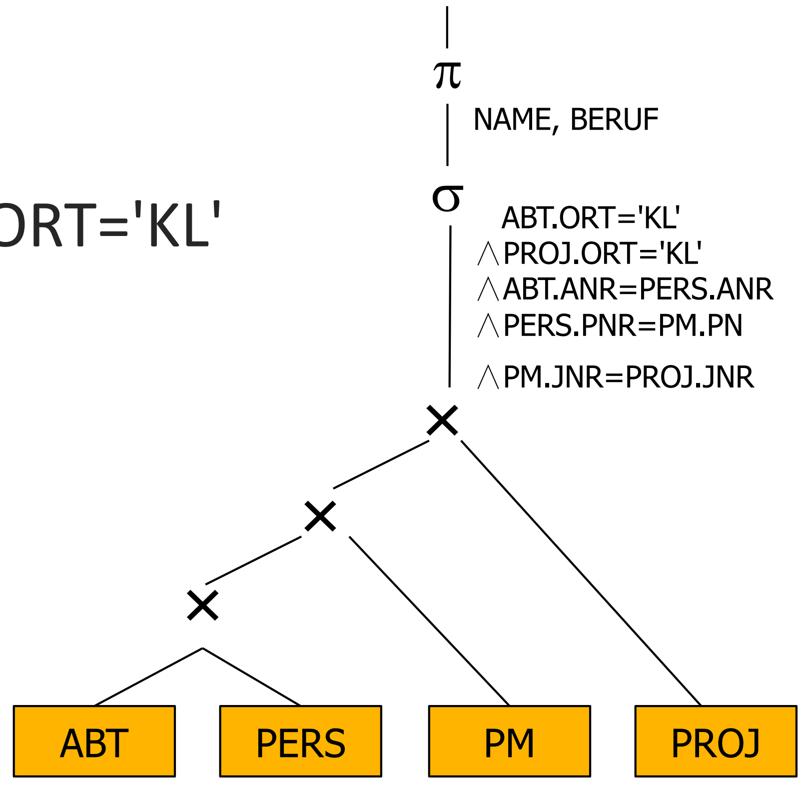
Sommersemester 2017

Technische Universität Kaiserslautern

Beispiel: SQL-Anfrage → Anfrageplan

- Beispiel: "Finde Name und Beruf von Angestellten, deren Abteilung in KL ist und die in KL Projekte durchführen"

```
SELECT PERS.NAME, PERS.BERUF
FROM ABT, PERS, PM, PROJ
WHERE ABT.ORT='KL' AND PROJ.ORT='KL'
AND ABT.ANR = PERS.ANR
AND PERS.PNR = PM.PN
AND PM.JNR = PROJ.JNR
```

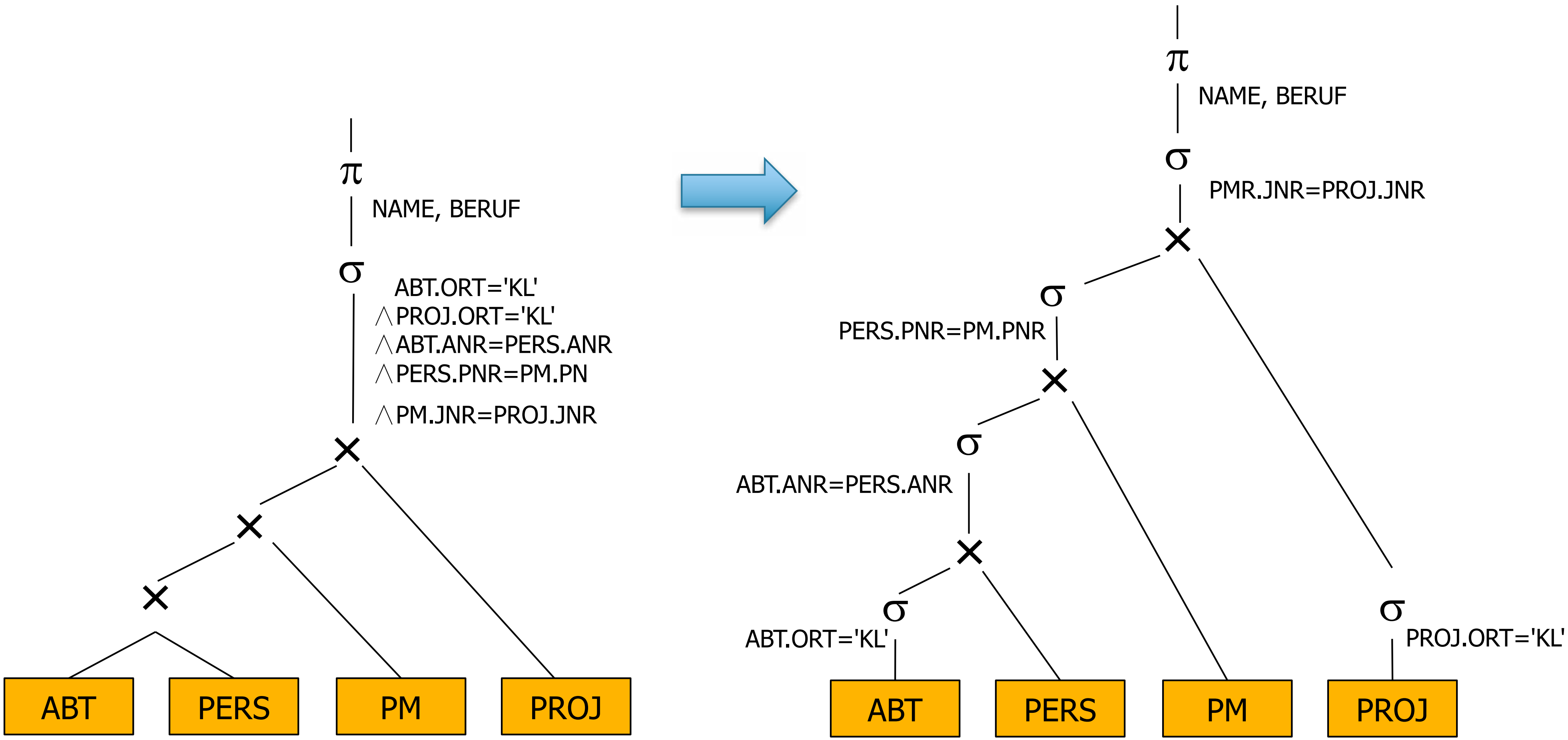


Relationenalgebra - Optimierung

- Relationenalgebraische Formulierungen spezifizieren Ausführungs-reihenfolge (prozedurale Elemente)
  - ➥ äquivalente Umformungen möglich
- Problem
  - gegeben: Ausdruck der Relationenalgebra (RA)
  - gesucht: äquivalenter, möglichst effizient auszuführender Ausdruck der Relationenalgebra
- Bestimmung einer möglichst guten Ausführungsreihenfolge (Einsatz von Heuristiken) für
  - unäre Operationen: π, σ
  - binäre Operationen: $\cap$, $\cup$, -, $\times$, $\bowtie$, $\div$
- Hier kommen die Äquivalenzregeln der relationalen Algebra zum Einsatz! (s. Kapitel 3)

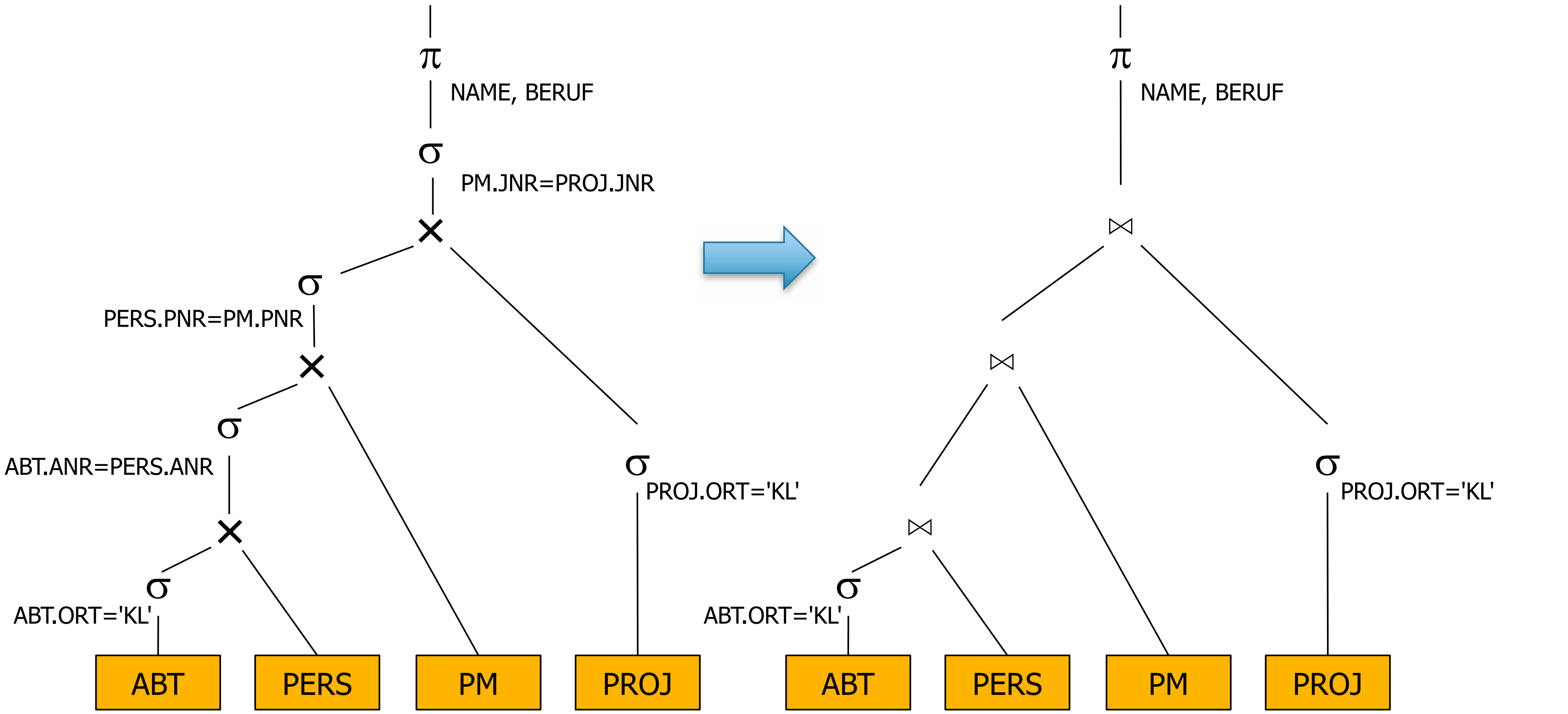
Heuristiken (1)

- Heuristik 1: Führe Selektionen so früh wie möglich aus
  - "Aufbrechen" von komplexen Selektionsbedingungen
  - Verschieben einzelner Selektionen in Richtung der Relationen



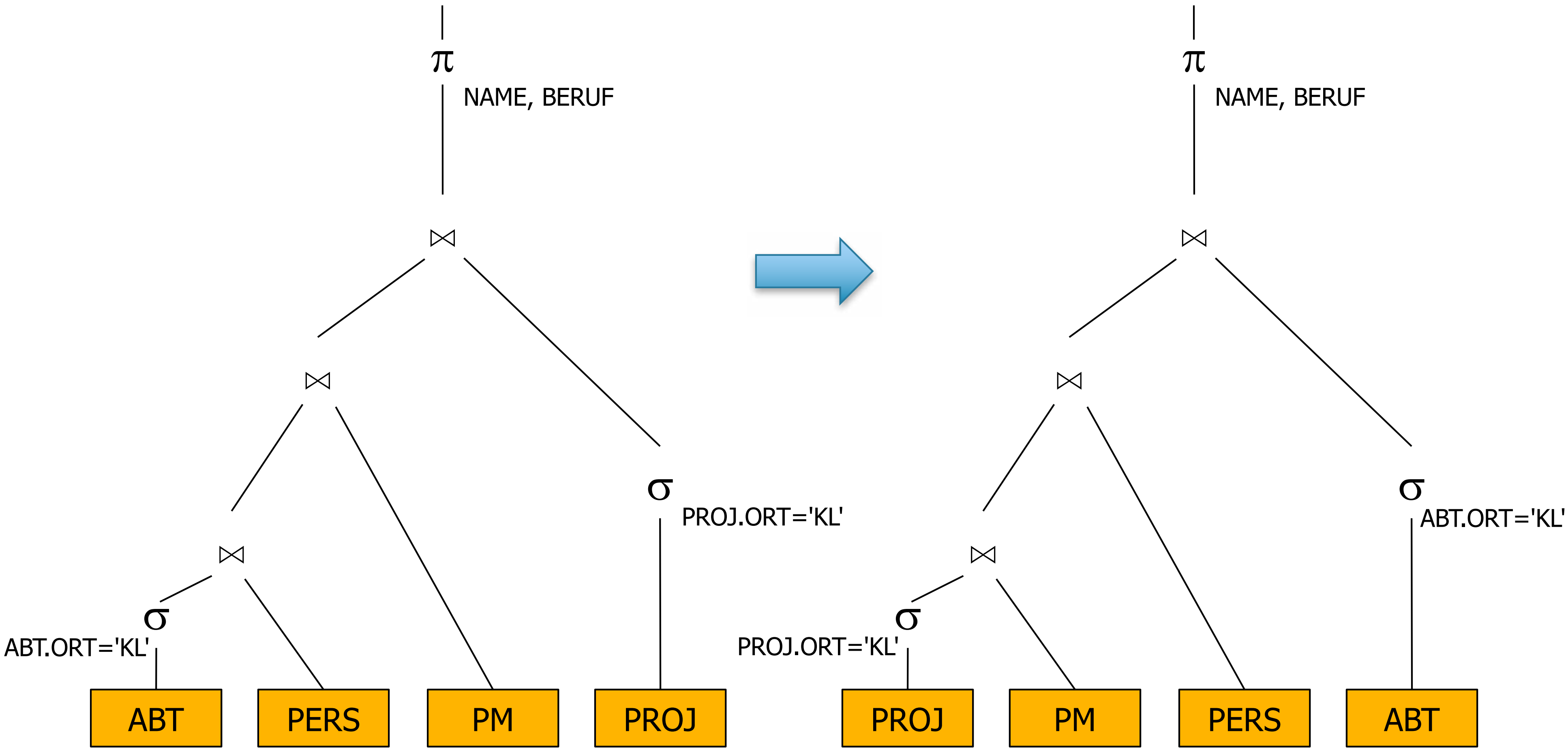
Heuristiken (2)

- Heuristik 2: Fasse Kreuzprodukt und Selektion zu Join zusammen
  - siehe Join-Definition: Produkt mit anschließender Selektion über den Eingaberelationen des Produkts



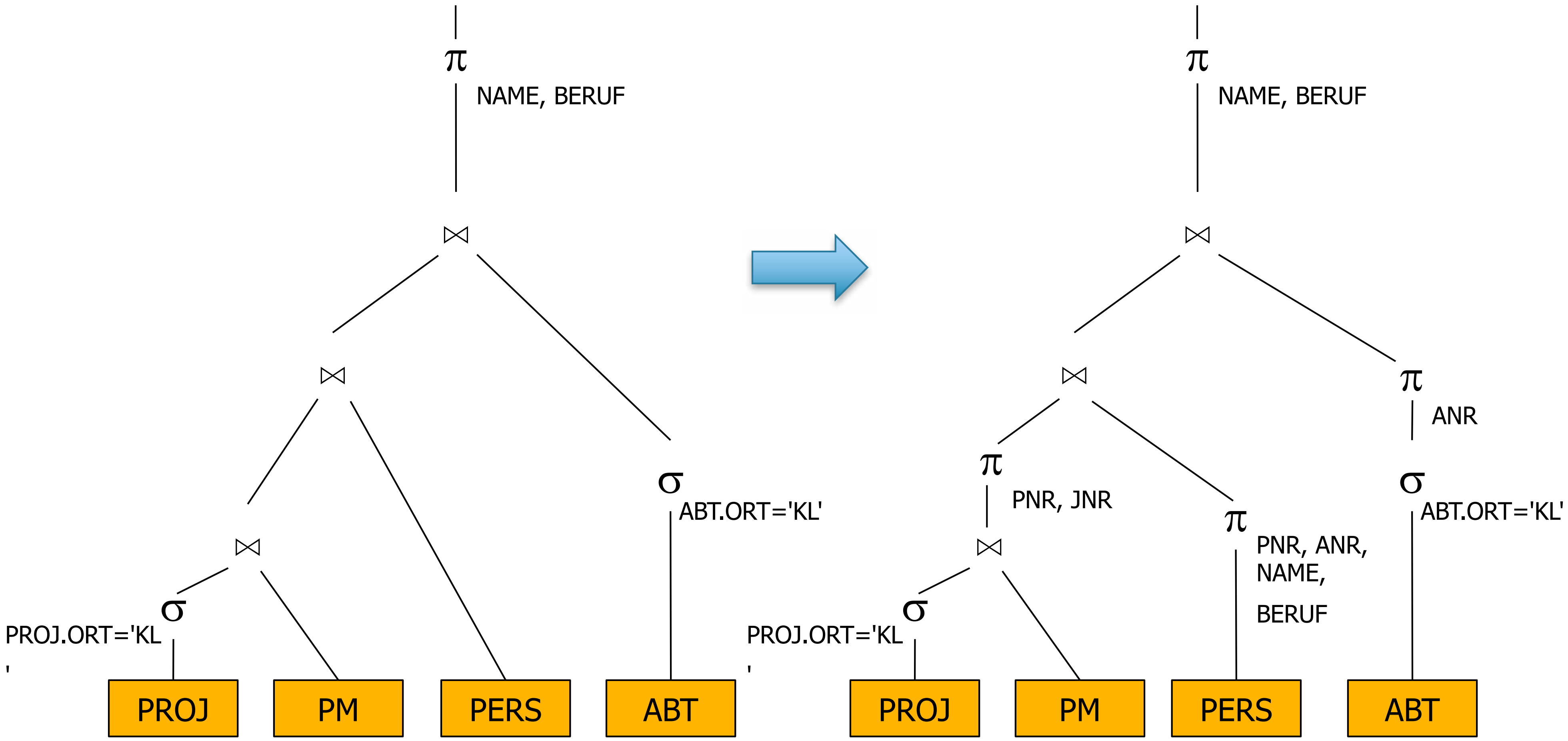
Heuristiken (3)

- Heuristik 3: Bestimme Join-Reihenfolge, sodaß möglichst kleine Zwischenresultate entstehen
  - erfordert Abschätzung der Resultatsgröße anhand der "Selektivität"
  - Vertauschen von Joins (Kommutativität, Assoziativität)



Heuristiken (4)

- Heuristik 4: Führe möglichst früh Projektionen (ohne Duplikateliminierung!) aus
  - Einfügen/Verschieben von Projektionsoperationen



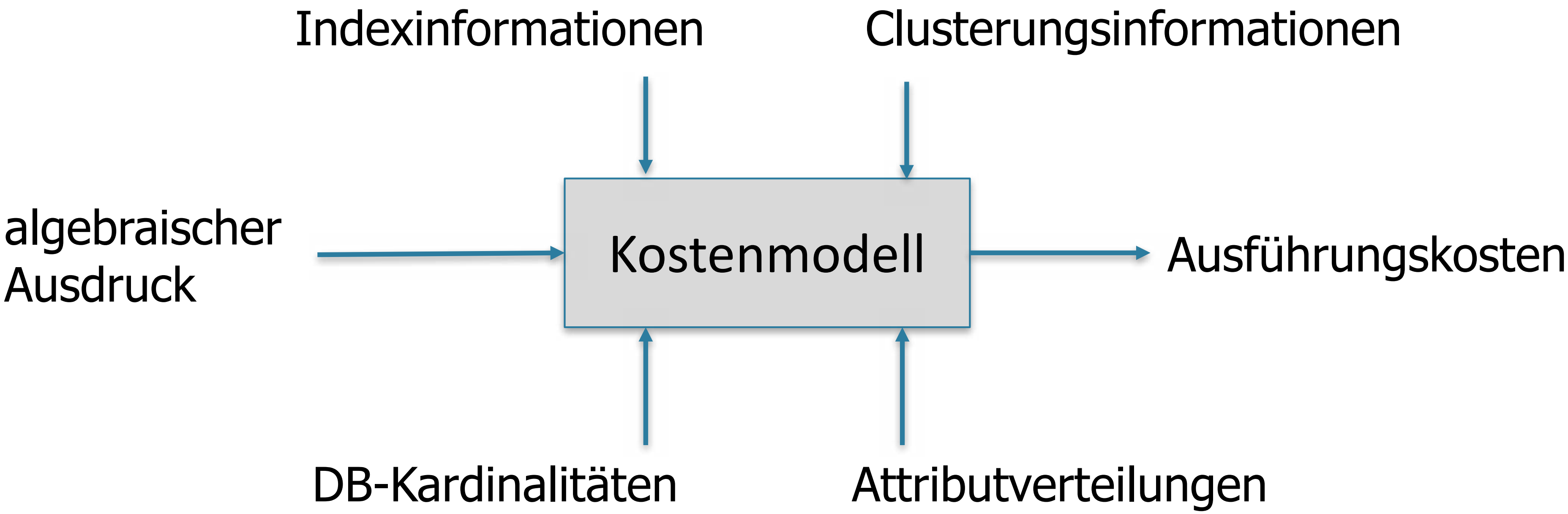
Heuristiken zur Algebraischen Optimierung

1. Führe Selektion so früh wie möglich aus!
2. Verknüpfe bestimmte Selektionen mit einem vorausgehenden Kartesischen Produkt zu einem Verbund!
3. Bestimme Verbundreihenfolge so, dass die Anzahl und Größe der Zwischenobjekte minimiert wird!
4. Führe Projektion (ohne Duplikateliminierung) frühzeitig aus!
5. Verknüpfe Folgen von unären Operationen wie Selektion und Projektion!
6. Fasse einfache Selektionen auf einer Relation zusammen!
7. Berechne gemeinsame Teilbäume nur einmal!
8. Verknüpfe bei Mengenoperationen immer zuerst die kleinsten Relationen!

Übersicht: Kostenbasierte Optimierung

- Generiere **alle** denkbaren Anfrageauswertungspläne:
  = Enumeration/Aufzählung möglicher Pläne
- Schätze deren Kosten ab:
  - Kostenmodell
  - Statistiken
  - Histogramme
  - Kalibrierung gemäß verwendetem Rechner
  - Abhängig vom verfügbaren Speicher
  - Aufwands-Kostenmodell
    - Durchsatz-maximierend
    - versus Antwortzeit-minimierend
- Behalte den Plan mit den geringsten geschätzten Kosten

Übersicht: Kostenmodelle



In dieser Vorlesung betrachten wir nur einfache Kosten in Form von Anzahl Zwischenergebnisse. Genauere Kosten hängen auch von Wahl der Implementierung der Operatoren, von verfügbaren Indexen, Performance der Hardware, etc. ab. → VL Datenbanksysteme.

Selektivitätsschätzung

- Selektivitätsschätzung: Idee
  - Gegeben: Anfrage und Relationen
  - Wie viele Tupel sind als Ergebnis zu erwarten?
  - Wie viele Tupel fallen als Zwischenergebnisse an?
- Selektivitätsschätzung: Werkzeuge
  - Kenntnisse/Statistiken über zugrunde liegende Daten
  - Generische Annahmen über Selektivität benutzter Prädikate
  - Schätzung der Selektivität von o.g. Operatoren (z.B., Join)

Selektivitätsschätzung: Grundlagen

- $T(R)$ ist die Anzahl der Tupel in Relation R
- $V(R, A)$ ist die Anzahl der verschiedenen Attributausprägungen für Attribut A.
- Dementsprechend für mehrere Attribute: $V(R, [A_1, A_2, \ldots A_n])$

- Aufgabe der Kostenschätzung
  - Gegeben eine Anfrage, wie groß ist das Ergebnis und wie viele Tupel fallen als Zwischenergebnisse an?
  - z.B. wie viele Tupel sind in $\sigma_{A=13434}(R)$
- Selektivität: Anteil der Tupel der Eingaberelation, die nicht herausgefiltert werden. Also: Größe der Ausgabe geteilt durch Größe der Eingabe.

Schätzungen für Selektion

- Gegeben eine Selektion $S = \sigma_{A=c}(R)$. Wie viele Tupel sind in S?
  - Schätzung:

$$T(S) = \frac{T(R)}{V(R, A)}$$

  - Gilt falls die Werte für A zufällig aus allen möglichen Werten gezogen wurden. (Gleichverteilung der Werte)
- Ungleichheit: Was ist bei $S = \sigma_{A<c}(R)$?
  - Im Allgemeinen, ohne weitere Annahmen: $T(S) = T(R)/2$
  - Aber, Intuition: man wählt mit solchen Bedingungen meist weniger Tupel aus.
  - Besser: $T(S) = T(R)/3$

Schätzung für "not-equals"

- Gegeben eine Selektion $S = \sigma_{A \neq c}(R)$. Wie viele Tupel sind in S?
- Einfache Schätzung
  - "Mehr oder weniger" alle Tupel erfüllen die Bedingung (naja, bis auf ein paar, aber egal)
  - Also: $T(S) = T(R)$
- Leicht verbessert
  - Die Tupel mit $A = c$ erfüllen die Bedingung nicht.
  - Also: $T(S) = T(R)\frac{V(R,A)-1}{V(R,A)}$
- Was passiert mit Kaskaden von Selektionen?

  Selektivität ist Produkt der einzelnen Selektivitäten!

Selektion mit ODER-Bedingungen

- Gegeben:

$$S = \sigma_{C_1 \vee C_2}(R)$$

- Annahme: die Bedingungen werden nie gemeinsam erfüllt
  - Also: entweder gilt C_1 oder C_2
  - Schätzung: Summe der beiden einzelnen Selektivitäten.
  - Beobachtung: Überschätzt oft. Was kann dann passieren?
- Besser: Annahme die Bedingungen sind unabhängig
  - Annahme: m_1 Tupel erfüllen C_1 und m_2 Tupel erfüllen C_2
  - Dann $T(S) = T(R) * (1 - (1 - \frac{m_1}{T(R)})(1 - \frac{m_2}{T(R)}))$

Selektion mit ODER-Bedingungen: Erläuterung

Wieso $T(S) = T(R) * (1 - (1 - \frac{m_1}{T(R)})(1 - \frac{m_2}{T(R)}))$?

$1 - \frac{m_1}{T(R)}$ ist der Anteil der Tupel, die C_1 **nicht** erfüllen; analog für C_2.

Dann gilt: $(1 - \frac{m_1}{T(R)})(1 - \frac{m_2}{T(R)})$ ist der Anteil der Tupel, die C_1 **und** C_2 **nicht erfüllen.**

$(1 - ***)$ ist der Anteil der Tupel, die C_1 **oder** C_2 erfüllen.

Klar: Multipliziert mit $T(R)$ ergibt $T(S)$

Andere Schätzer

- Projektion
  - Ändert die Kardinalität nicht (unter Bag-Semantik)
  - Sehr wohl aber die Größe der Tupel!
  - Bei Mengensemantik: $T(S) = V(R, A)$ mit $S = \pi_A(R)$
- Kartesisches Produkt
  - Einfach: Produkt der beteiligten Kardinalitäten

Jetzt wird es **unklar was zu tun** ist:

- Union
  - Obere Schranke: Summe der beiden Kardinalitäten
  - Untere Schranke: Größere der beiden Kardinalitäten
  - Empfehlung aus Literatur: Irgendwas dazwischen, z.B. größere plus die halbe kleinere Kardinalität

Andere Schätzer

- Schnitt
  - Obere Schranke: Kleinste der beiden Kardinalitäten
  - Untere Schranke: 0
  - Empfehlung aus Literatur: Durchschnitt dieser beiden Werte.
- Differenz $R - S$
  - Obere Schranke: $T(R)$
  - Untere Schranke: $T(R) - T(S)$
  - Empfehlung aus Literatur: $T(R) - \frac{T(S)}{2}$

Schätzung für Joins: Natürlicher Join

- Zwei Relationen: $R = (X, Y)$ und $S = (Y, Z)$
  Annahme: Y ist ein einfaches Attribut, keine Menge von Attributen.
  X und Z dürfen Mengen sein.
- Was können wir dann sagen? Leider nur sehr wenig, da z.B. folgende Fälle auftreten können:
  - Die beiden Relationen haben disjunkte Mengen für Y-Werte. Also ist der Join leer, d.h. $T(R \bowtie S) = 0$.
  - Y ist Schlüssel von S und Fremdschlüssel in R. Dann findet jedes Tupel in R einen Joinpartner in S, also $T(R \bowtie S) = T(R)$.
  - Fast alle Tupel aus R und S haben den gleichen Wert für Y, also $T(R \bowtie S) = T(R) * T(S)$.

Schätzung für Joins: Natürlicher Join: Annahmen

- Häufig auftretende Fälle. Weitere Annahmen:
  - Es gibt Y-Werte $y_1, y_2, y_3, \ldots$
  - Relationen benutzen diese Werte in dieser Reihenfolge.
  - Dann: Falls $V(R, Y) \leq V(S, Y)$ so gilt auch, dass jeder Y-Wert in R auch ein Y-Wert in S ist.
- Erhaltung der Wertemengen
  - Falls A kein Join-Attribut ist, gilt
    $$V(R \bowtie S, A) = V(R, A)$$
  - D.h. Attribut A verliert durch den Join keine möglichen Werte.

Schätzung für Joins: Natürlicher Join

- Gesucht: Größe des Joins $R(X,Y) \bowtie S(Y,Z)$
  - Gegeben, zwei Tupel: $r \in R$ und $s \in S$
  - Was ist die Wahrscheinlichkeit, dass $r.Y = s.Y$?
  - Annahme: $V(R,Y) \geq V(S,Y)$,
    also gibt es den Y-Wert von s in R.
  - Also: Wahrscheinlichkeit, dass $r.Y = s.Y$ ist $1/V(R,Y)$
  - Umgekehrt: falls $V(R,Y) < V(S,Y)$ analog.
  - Insgesamt: Übereinstimmung in Y mit Wahrscheinlichkeit $1/\max(V(R,Y),V(S,Y))$

$$T(R \bowtie S) = \frac{T(R) * T(S)}{\max(V(R,Y),V(S,Y))}$$

Wie können Größen abgeschätzt werden?

- Wichtig: Statistiken über Werte von Attributen, Größen von Relationen. Aber wie kann man diese berechnen und repräsentieren?
- Durch Scannen der gesamten Mengen
  - Kann hin und wieder berechnet werden, natürlich besser nicht zur Anfragezeit.
  - Moderne DMBS haben bestimmte Befehle dafür.
- Ausprägungen für Attribut A
  - Falls $V(R, A)$ nicht zu groß ist: speichere einen Zähler für jeden Wert von A. Z.B. es gibt 50 Tupel in Relation Professoren mit Rang='W3'.
  - Sonst: Gruppierung. Evtl: Exaktes Speichern für eine Teilmenge der Werte (z.B. der häufigsten).

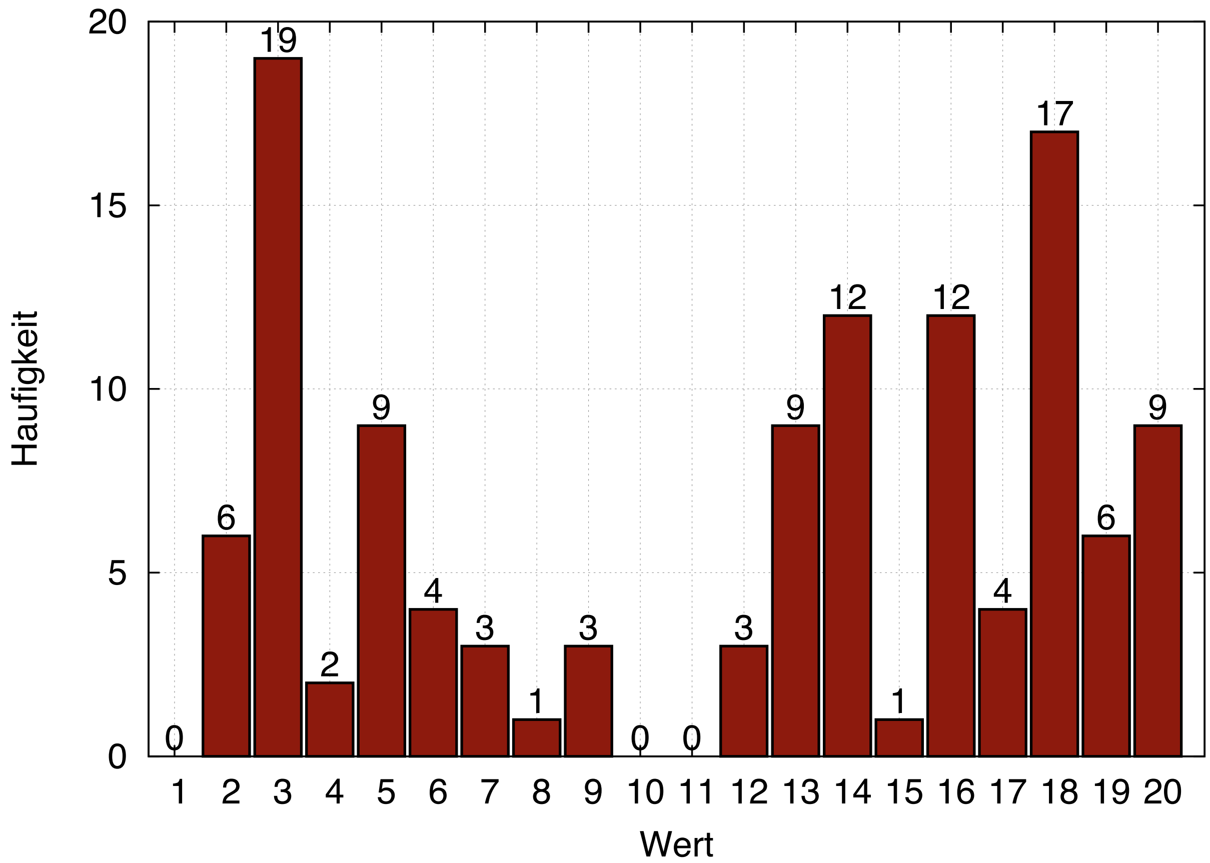
→ Histogramme oder parametrisierte Verteilungen!

Beispieldaten

{2, 5, 3, 20, 18, 7, 16, 18, 18, 2, 2, 17, 14, 3, 20, 3, 16, 6, 7, 3, 16, 16, 15, 5, 20, 13, 16, 20, 12, 14, 13, 3, 14, 18, 14, 14, 16, 18, 19, 3, 5, 2, 5, 14, 20, 17, 3, 17, 16, 3, 2, 19, 3, 9, 13, 4, 3, 16, 14, 13, 13, 16, 20, 14, 4, 2, 3, 18, 7, 3, 5, 3, 6, 9, 18, 3, 16, 18, 20, 18, 5, 18, 5, 18, 13, 14, 19, 13, 14, 3, 14, 18, 14, 18, 18, 16, 19, 5, 3, 17, 18, 3, 19, 3, 20, 9, 16, 12, 20, 8, 12, 13, 13, 19, 18, 6, 3, 5, 18, 6}

Verteilung der Daten

- Wert → Häufigkeit.
- Dargestellt im "Histogramm-Stil":



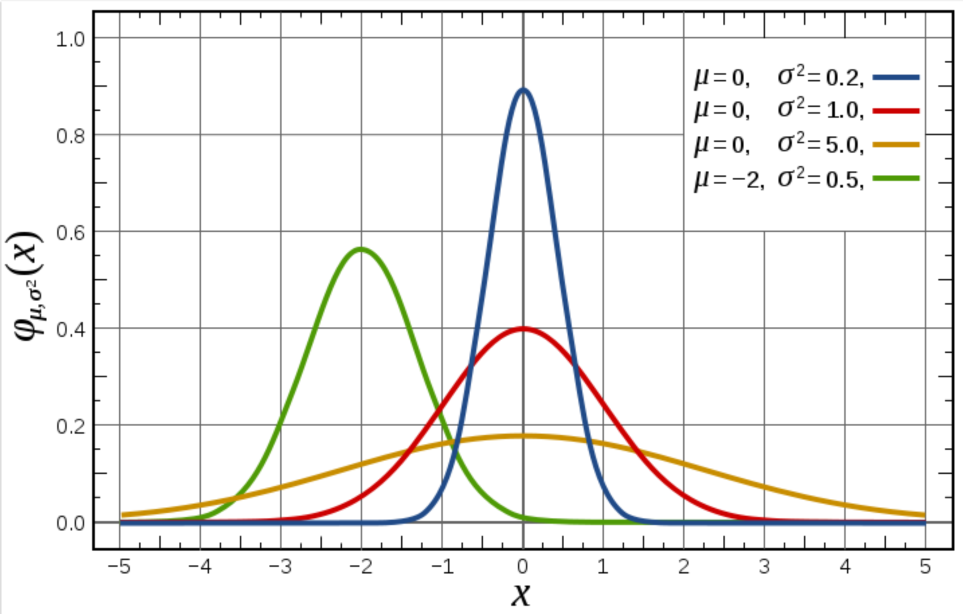
- Wir sehen: Es liegen 20 Werte im Wertebereich. Es gibt 120 Datenpunkte.
- Alternativ: nicht die absolute Häufigkeit sondern normalisiert in [0,1], also "Wahrscheinlichkeit" bei zufälligem Zugriff auf Daten einen bestimmten Datenpunkt zu treffen.

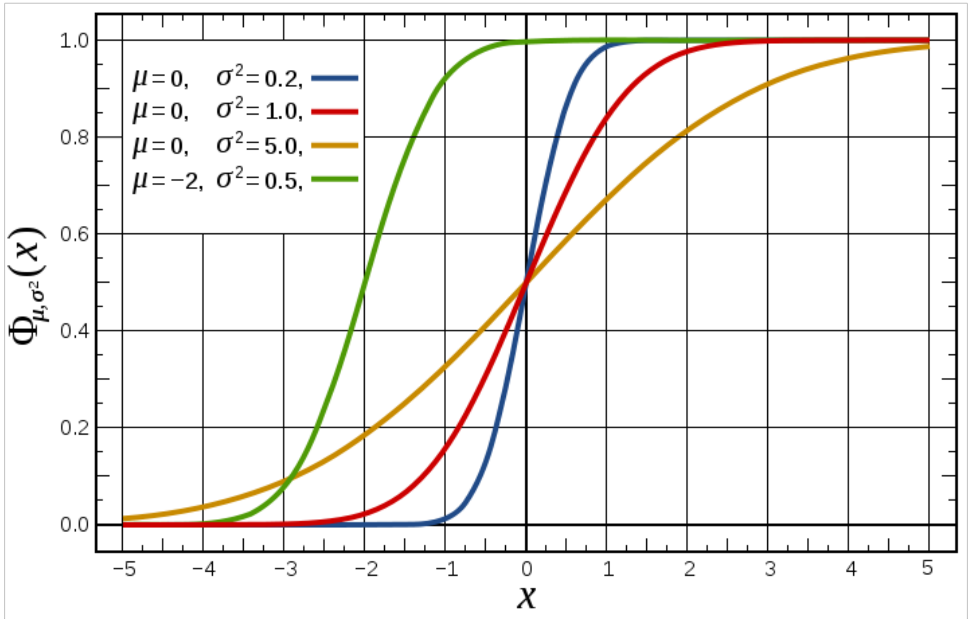
Zusammenfassen/Beschreiben großer Datenmengen

- Beschreibung durch parametrisierte Verteilung (Funktion)
  - Z.B. die Daten folgen einer Normalverteilung mit Erwartungswert μ und Varianz σ^2.
- Zusammenfassen von Werten in Zellen (aka. Eimer oder Buckets)
  - Wie viele Werte fallen in [0,5[
    wie viele Werte fallen in [5,10[
    etc.

Parametrisierte Verteilungen

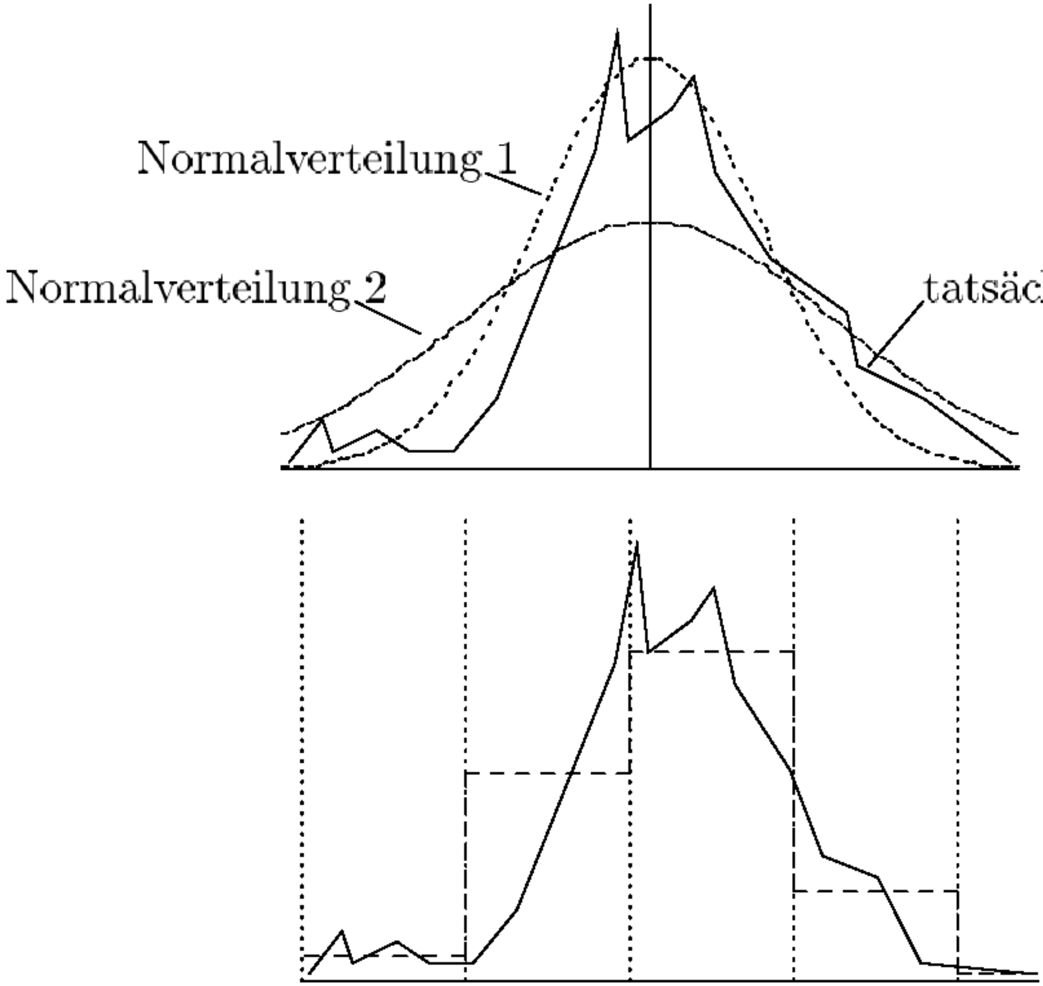
- Dichtefunktion und Verteilungsfunktion der Normalverteilung:





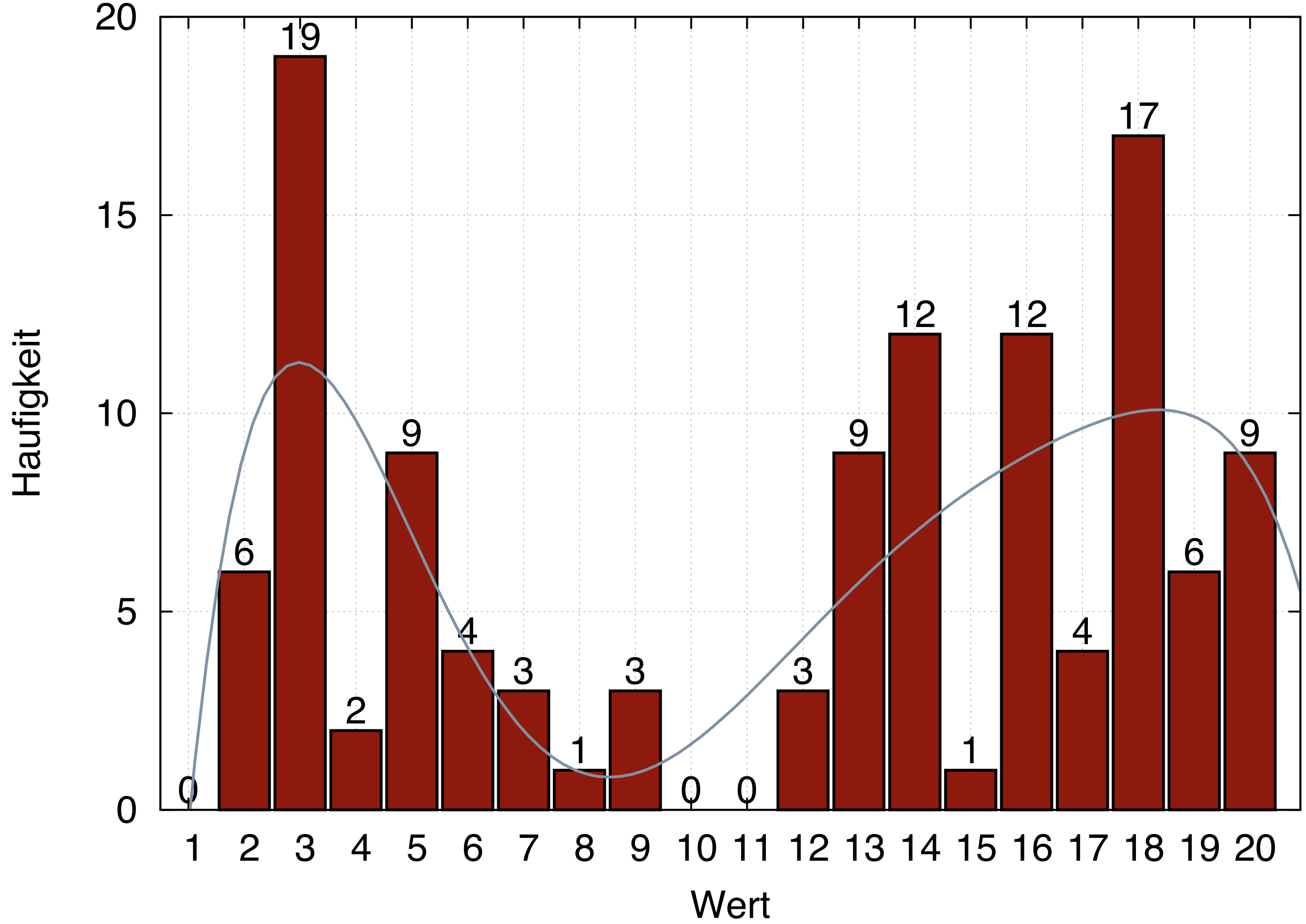
Abbildungen aus Wikipedia

Parametrisierte Verteilungen und Histogramme



Oft ist es schwierig eine tatsächliche Verteilung durch eine parametrisierte Verteilung auszudrücken. Histogramme sind flexibler.

Parametrisierte Verteilungen

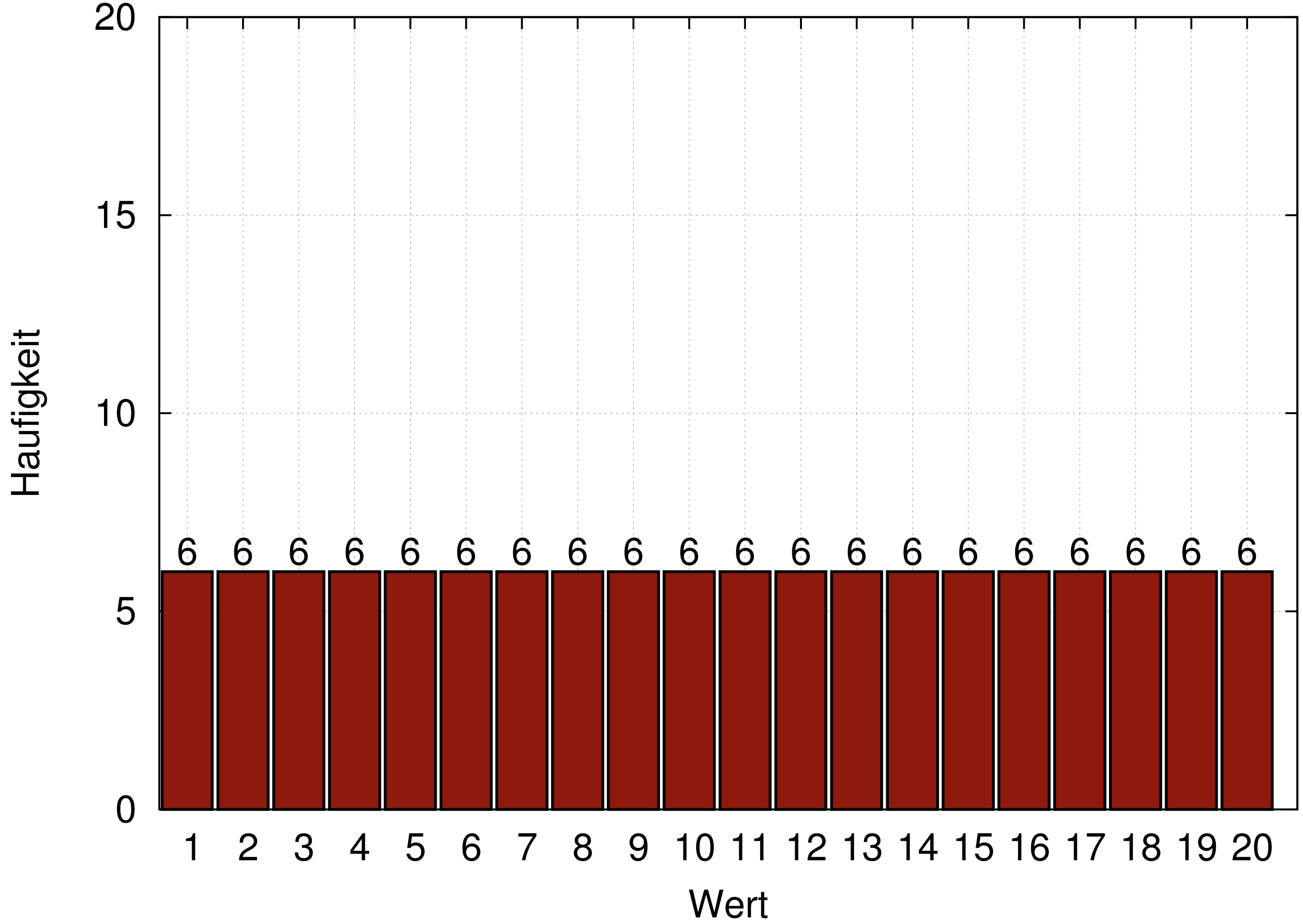


Hier, fit durch Polynom 6. Grades (mit Hilfe des Tools xmgrace):

$$f(x) := -21.884 + 29.533 * x - 9.2268 * x^2 + 1.2529 * x^3 \\ -0.085019 * x^4 + 0.0028667 * x^5 - 3.8485 * 10^{-5} * x^6$$

Annahme Gleichverteilung

Es liegen 20 Werte im Wertebereich. Es gibt 120 Datenpunkte. Jeder Wert kommt, unter Annahme einer Gleichverteilung, also 6 Mal vor.

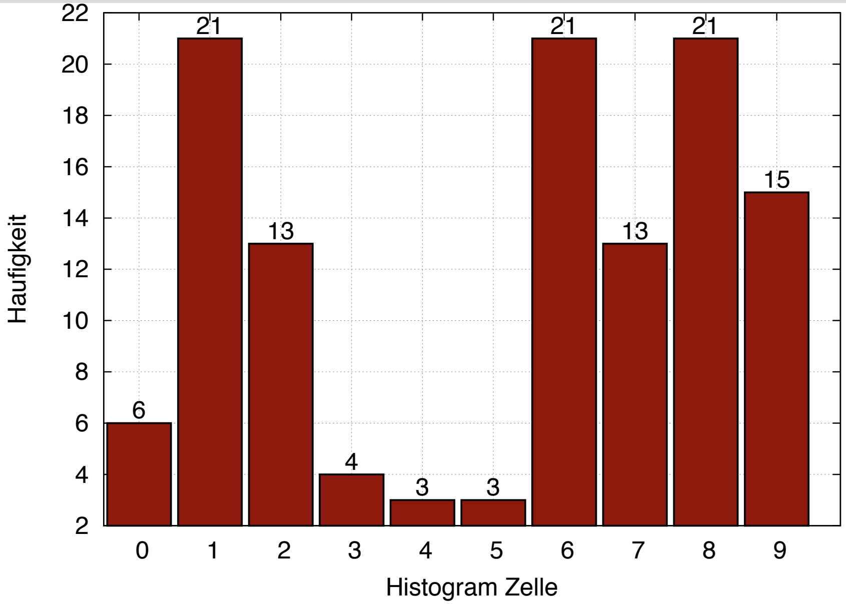


Oft nicht sehr realistische Darstellung (aber äußerst kompakt).

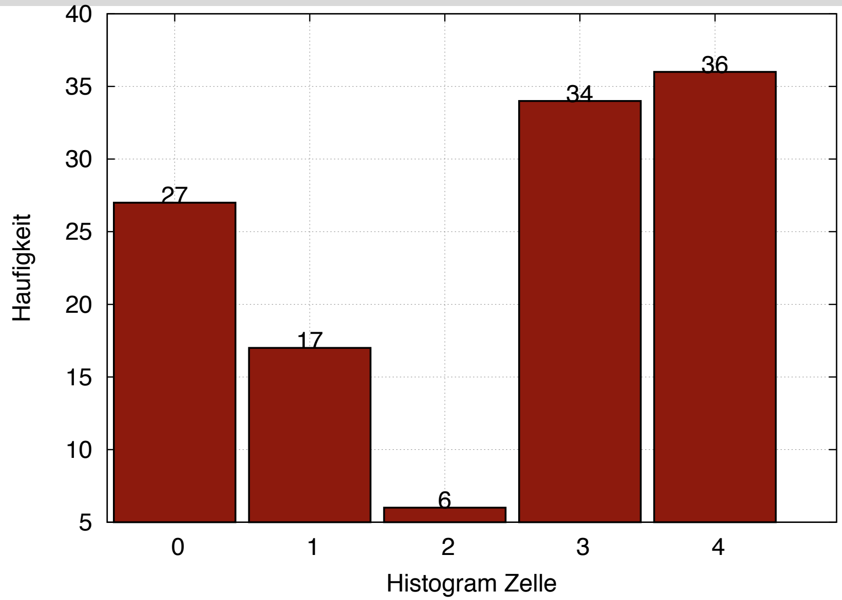
Histogramme

- Histogramm teilt den Wertebereich in Zellen oder Intervalle (Englisch: buckets oder cells). Für jede dieser Zellen wird die Anzahl der Elemente gespeichert, die in die Zelle fallen.
- Equi-Width-Histogramme:
  - Zellen habe immer die gleiche Breite im Wertebereich
- Equi-Depth (oder equi-height genannt) Histogramme:
  - Zellen haben die gleiche "Höhe"
  - Um dies zu erreichen: Breite der Zellen wird angepasst

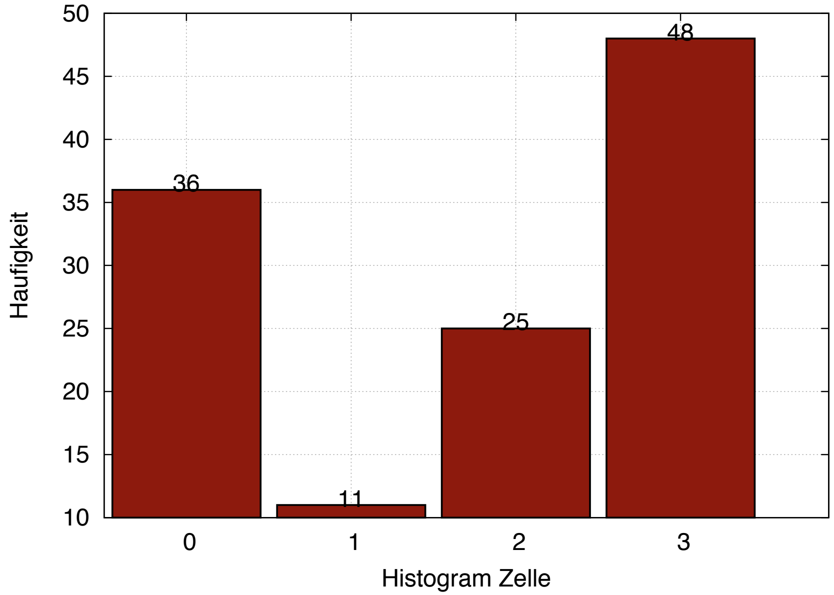
Equi-Width-Histogramme (Beispiele an o.g. Daten)



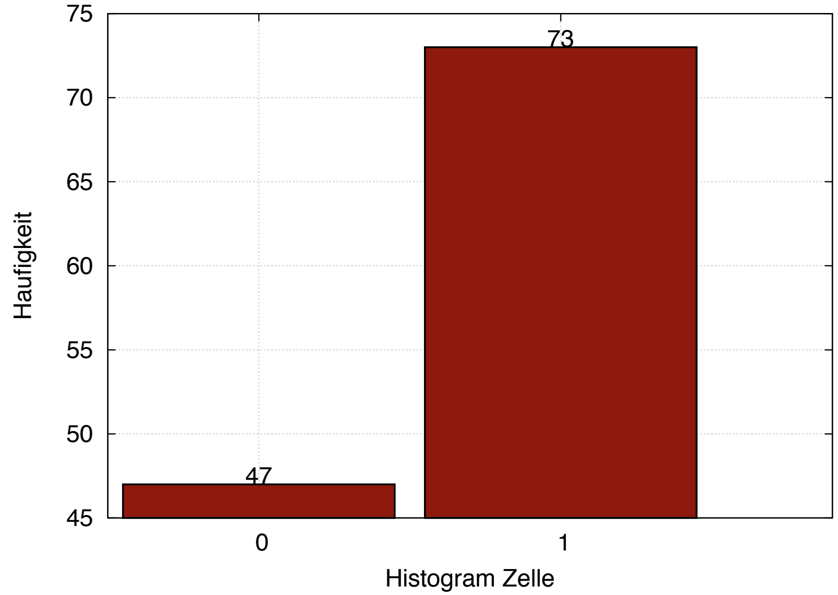
(a) Width = 2



(b) Width = 4

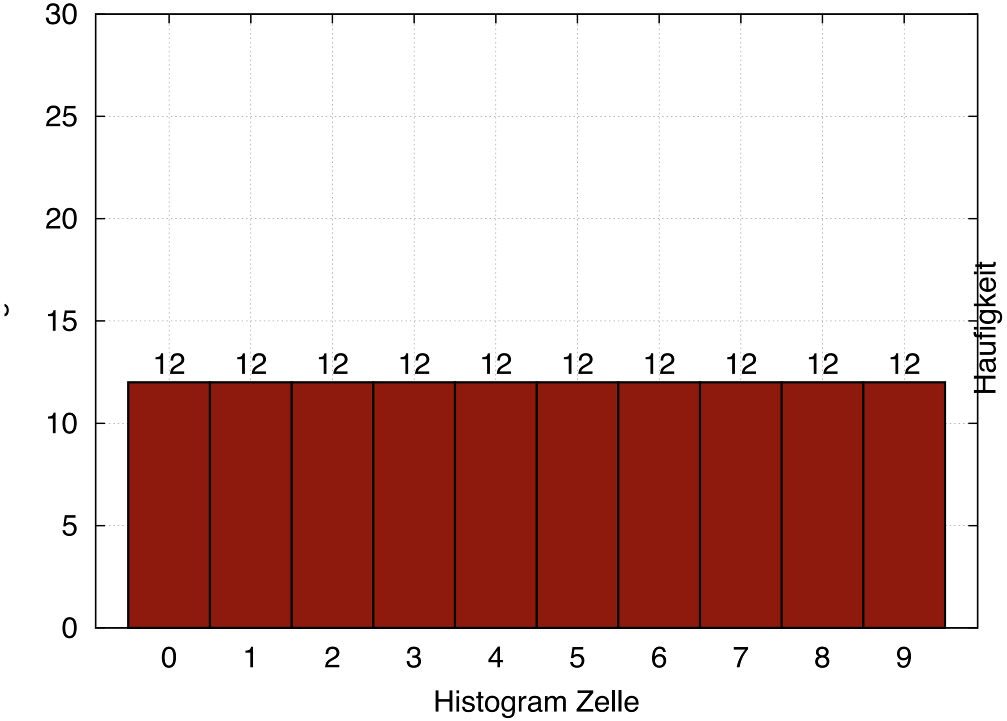


(c) Width = 5

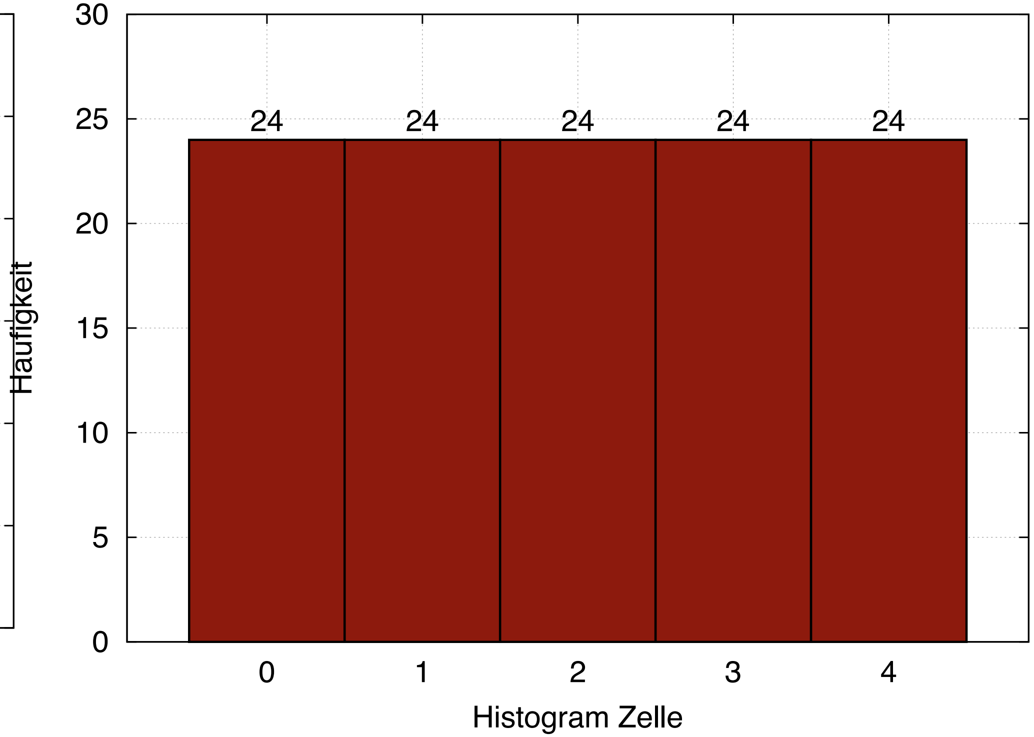


(d) Width = 10

Equi-Depth-Histogramme (Beispiele an o.g. Daten)



(a) Depth = 10%



(b) Depth = 20%

- Grenzen der Zellen (je rechter Punkt, d.h. Ende)
  a) 3, 3, 5, 12, 14, 15, 16, 18, 19, 20
  b) 3, 12, 15, 18, 20

Punktanfragen und Bereichsanfragen auf Histogrammen

- Punktanfragen
  - Wie viele Tupel haben den Wert A = 10?
  - Nachschauen: Zelle in die der Wert 10 fällt.
  - Annahme hier: Gleichverteilung innerhalb der Zelle.
  - Resultat: Anzahl der Tupel geteilt durch Breite der Zelle.
  - Bzw. wenn bereits "normalisiert" dann nur Wert der Zelle.
- Bereichsanfragen
  - Wie viele Tupel haben einen Wert A > 10?
  - Nachschauen: In welche Zelle fällt der Wert 10? (Achtung insbesondere bei Equi-Depth Histogrammen)
  - Resultat: Summe der Größen der darüber liegenden Zellen und anteilsmäßig diese Zelle (wie oben).
  - Oder: Histogramm für kumulative Verteilung betrachten

Schätzung mit Histogrammen

- Obwohl man für diskrete Daten auch Punktanfragen bearbeiten kann, ist dies im kontinuierlichen Fall nicht möglich.
- Es liegen unendlich viele Werte in jeder Histogrammzelle (mit Häufigkeit 0)
- D.h. es kann im kontinuierlichen Fall "nur" nach Häufigkeiten für Intervalle gefragt werden.

▪ Fehlermaße

- Ist für einen exakten Wert x eine Schätzung (Näherungswert) $\hat{x}$ gegeben,
  - so heisst $|\hat{x} - x|$ absoluter Fehler und
  - $\frac{|\hat{x}-x|}{x}$ im Fall $x \neq 0$ relativer Fehler
- Für mehrere solcher Beobachtungen: Sum Squared Error (SSE), Mean Absolute Error (MAE), Mean Squared Error (MSE)

Zusammenfassung Anfrageoptimierung

- Deklarative Anfrage in Form von SQL wird vom Datenbanksystem in konkreten Anfrageplan überführt.
- Reihenfolge/Ordnung der Operatoren (der rel. Algebra) des Anfrageplans können durch Regeln verändert werden.
- Haben hier einfache regelbasierte Anfrageoptimierung betrachtet: Richtlinien wie Operatoren im Baum generell angeordnet sein sollten.
- Ebenfalls betrachtet: Einfaches Kostenmodell, mit dessen Hilfe Kosten (=Anzahl Zwischenergebnisse) abgeschätzt werden können.
- Bessere Abschätzung anhand konkreter Verteilungsmodelle.
- Ausblick Vorlesung Datenbanksysteme: Mehr zu Kostenschätzung, Optimierung der Join-Reihenfolge, verschiedene Histogramme, Abschätzung von Anzahl Unique Values durch "Probabilistic Counting", Index-Tuning, Schema Denormalisierung.