

**Dieses Dokument ist eine Zweitveröffentlichung (Postprint) /**

**This is a self-archiving document (accepted version):**

Philipp Rösch, Wolfgang Lehner

## **Sample Footprints für Data-Warehouse-Datenbanken**

**Erstveröffentlichung in / First published in:**

*Computer Science - Research and Development*. 2010, 25, S. 217–233 [Zugriff am: 20.01.2023]. Springer. ISSN 1865-2042.

DOI: <http://dx.doi.org/10.1007/s00450-009-0100-x>

Diese Version ist verfügbar / This version is available on:

<https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa2-830138>

# Sample Footprints für Data-Warehouse-Datenbanken

Philipp Rösch · Wolfgang Lehner

**Zusammenfassung** Durch stetig wachsende Datenmengen in aktuellen Data-Warehouse-Datenbanken erlangen Stichproben eine immer größer werdende Bedeutung. Insbesondere interaktive Analysen können von den signifikant kürzeren Antwortzeiten der approximativen Anfrageverarbeitung erheblich profitieren. Linked-Bernoulli-Synopsen bieten in diesem Szenario speichereffiziente, schemaweite Synopsen, d. h. Synopsen mit Stichproben jeder im Schema enthaltenen Tabelle bei minimalem Mehraufwand für die Erhaltung der referenziellen Integrität innerhalb der Synopse. Dies ermöglicht eine effiziente Unterstützung der näherungsweise Beantwortung von Anfragen mit beliebigen Fremdschlüsselverbundoperationen.

In diesem Artikel wird der Einsatz von Linked-Bernoulli-Synopsen in Data-Warehouse-Umgebungen detaillierter analysiert. Dies beinhaltet zum einen die Konstruktion speicherplatzbeschränkter, schemaweiter Synopsen, wobei unter anderem folgende Fragen adressiert werden: Wie kann der verfügbare Speicherplatz auf die einzelnen Stichproben aufgeteilt werden? Was sind die Auswirkungen auf den Mehraufwand? Zum anderen wird untersucht, wie Linked-Bernoulli-Synopsen für die Verwendung in Data-Warehouse-Datenbanken angepasst werden können. Hierfür werden eine inkrementelle Wartungsstrategie sowie eine Erweiterung um eine Ausreißerbehandlung für die Reduzierung von Schätzfehlern approximativer Antworten von Aggregationsanfragen mit Fremdschlüsselverbundoperationen vorgestellt.

---

P. Rösch (✉) · W. Lehner  
Fakultät Informatik, Institut für Systemarchitektur,  
Lehrstuhl für Datenbanken, Technische Universität Dresden,  
01062 Dresden, Deutschland  
e-mail: philipp.roesch@tu-dresden.de

W. Lehner  
e-mail: wolfgang.lehner@tu-dresden.de

Eine Vielzahl von Experimenten zeigt, dass Linked-Bernoulli-Synopsen und die in diesem Artikel präsentierten Verfahren vielversprechend für den Einsatz in Data-Warehouse-Datenbanken sind.

**Schlüsselwörter** Data-Warehouse-Datenbanken · Approximative Anfrageverarbeitung · Bernoulli-Stichproben · Ausreißerer kennende Synopsen

**Abstract** With the amount of data in current data warehouse databases growing steadily, random sampling is continuously gaining in importance. In particular, interactive analyses of large datasets can greatly benefit from the significantly shorter response times of approximate query processing. In this scenario, Linked Bernoulli Synopses provide memory-efficient schema-level synopses, i. e., synopses that consist of random samples of each table in the schema with minimal overhead for retaining foreign-key integrity within the synopsis. This provides efficient support to the approximate answering of queries with arbitrary foreign-key joins. In this article, we focus on the application of Linked Bernoulli Synopses in data warehouse environments. On the one hand, we analyze the instantiation of memory-bounded synopses. Among others, we address the following questions: How can the given space be partitioned among the individual samples? What is the impact on the overhead? On the other hand, we consider further adaptations of Linked Bernoulli Synopses for usage in data warehouse databases. We show how synopses can incrementally be kept up-to-date when the underlying data changes. Further, we suggest additional outlier handling methods to reduce the estimation error of approximate answers of aggregation queries with foreign-key joins. With a variety of experiments, we show that Linked Bernoulli Synopses and the proposed tech-

niques have great potential in the context of data warehouse databases.

**Keywords** Data Warehouse Databases · Approximate Query Processing · Bernoulli Sampling · Outlier-Aware Sample Synopses

**CR subject classification** H.2.4 · H.2.7 · H.3.3 · H.4.2

## 1 Einleitung

Der stetig zunehmende Einsatz von Sensoren, angefangen von analogen Erfassungseinheiten bis hin zu RFIDs, erzeugt eine Datenflut, die nicht nur zu speichern, sondern auch mit Hilfe statistischer Analyse auszuwerten ist, so dass sinnhafte und relevante Informationen aus diesen Datenbeständen abgeleitet werden können. Wesentliches Charakteristikum ist hierbei, dass nicht vorgefertigte Berichte zu erstellen sind, sondern interaktive Analysen – von der explorativen Analyse multidimensionaler Datenwürfel bis hin zur Ad-Hoc-Berechnung von Data Mining-Modellen – gefordert werden. Für die Ebene der Datenbanksysteme sind kurze Anfrageverarbeitungszeiten hierfür essenziell. Mit stetig wachsender Datenmenge in Data-Warehouse-Umgebungen als Ort der zentralen Bereitstellung aufbereiteter und zur Analyse freigegebener Datenbasen wird dies jedoch immer schwieriger zu bewältigen. Als Lösung bieten sich eine Vielzahl unterschiedlicher Techniken und Methoden an, die sich von der massiven Skalierung auf Ebene der Hardware („scale up/scale out“) bis hin zu spezifischen Datenstrukturen innerhalb des Datenbanksystems erstrecken. In diesem Beitrag wird die Idee verfolgt, dass in dem Kontext der Massendatenauswertung (zumindest in einem ersten Analyseschritt) nicht immer eine korrekte Antwort basierend auf den Basisdaten notwendig ist, sondern sich die Analysen auch mit einer Stichprobe und einer darauf aufsetzenden näherungsweise Anfrageverarbeitung begnügen – kleine Synopsen der Basisdaten sind daher von großer Bedeutung. Stichproben zeichnen sich dabei gegenüber anderen Synopsenformen durch ihre einfache Verwendung sowie die Anwendbarkeit auf ein breites Anfragespektrum aus. Darüber hinaus bieten sie probabilistische Fehlergrenzen.

Im Datenbankenbereich wurden verschiedene Stichprobenverfahren für verschiedene Anfragetypen und Anwendungsszenarien vorgestellt. Diese beschränken sich jedoch größtenteils auf einzelne Tabellen. Infolgedessen sind diese Stichprobenverfahren nicht für die näherungsweise Beantwortung von Verbundanfragen geeignet. Wie in [1, 9] gezeigt wird, kann das Verbundergebnis einfacher Zufallsstichproben sehr klein und verzerrt sein, d. h. im Allgemeinen führt der Verbund von Stichproben nicht zu einer Stichprobe des Verbundes der entsprechenden Tabellen.

In diesem Beitrag werden die Betrachtungen auf Fremdschlüsselverbunde – die üblichen Verbundoperationen in Data-Warehouse-Szenarien – beschränkt. Hierfür stellten Acharya et al. *Join-Synopsen* [1] vor, welche die referenzielle Integrität erhalten und somit die näherungsweise Beantwortung von Anfragen mit Fremdschlüsselverbunden ermöglichen. Dieser Ansatz hat die Anwendbarkeit von Synopsen in Data-Warehouse-Umgebungen deutlich gesteigert. Jedoch ist die Speicherausnutzung von Join-Synopsen nicht effektiv: Viele Tupel werden nur für die Wahrung der referenziellen Integrität gespeichert und können nicht für die näherungsweise Anfrageverarbeitung genutzt werden. *Linked-Bernoulli-Synopsen* [14] erweitern Join-Synopsen, indem sie den Zufall der Stichprobenerhebung von über Fremdschlüsselbeziehungen verbundenen Tabellen wiederverwenden. Die grundlegende Idee ist dabei, die Aufnahme eines Tupel in die Stichprobe davon abhängig zu machen, ob dieses Tupel von einem anderen Tupel referenziert wird, welches bereits in die entsprechende Stichprobe aufgenommen wurde. Dadurch wird die Anzahl der nur für die Wahrung der referenziellen Integrität gespeicherten Tupel signifikant gesenkt, oder genauer gesagt, Linked-Bernoulli-Synopsen minimieren den Speicherbedarf der Synopsen.

Hauptgegenstand dieses Artikels ist der praktische Einsatz von Linked-Bernoulli-Synopsen. Nach einer kurzen Vorstellung des zu Grunde liegenden Prinzips in Abschn. 2 wird das Problem der Synopsenberechnung für gegebene Speicherplatzbeschränkungen analysiert. Dazu werden in Abschn. 3 verschiedene *Speicherplatzaufteilungsstrategien* vorgestellt. Die Schwierigkeit dabei entsteht durch die Erhaltung der referenziellen Integrität: Die Veränderung einer Stichprobengröße kann Auswirkungen auf die Größe vieler anderer Stichproben haben. Infolgedessen muss für jede Parameteränderung die resultierende Synopsengröße neu berechnet werden; dies jedoch erfordert das Lesen sämtlicher Tabellen. Um die Synopsenberechnung dennoch effizient zu gestalten, werden in Abschn. 4 verschiedene Möglichkeiten der *Abschätzung der Synopsengröße* präsentiert. Anschließend wird in Abschn. 5 der Fokus von der Synopsenerstellung auf die Verwendung von Linked-Bernoulli-Synopsen in Data-Warehouse-Szenarien gerichtet. Dabei werden die Problematik der *Synopsenwartung* aufgrund von Einbringungen neuer Datenbestände in die Faktentabelle (Abschn. 5.1) sowie eine Erweiterung um eine *Optimierung für Aggregationsanfragen* für eine zusätzliche Reduzierung des Schätzfehlers (Abschn. 5.2) diskutiert. Eine ausführliche Evaluation der vorgestellten Verfahren ist in Abschn. 6 zu finden.

## 2 Linked-Bernoulli-Synopsen

Als Grundlage für die folgenden Diskussionen dieses Artikels werden in diesem Abschnitt der Aufbau und die

Idee von Linked-Bernoulli-Synopsen vorgestellt und der Ablauf des Stichprobenerhebungsprozesses überblicksartig beschrieben.

## 2.1 Grundlagen

Sei  $G$  ein Schemagraph einer relationalen Datenbank mit den Tabellen der Datenbank als Knoten; die Fremdschlüsselbeziehungen zwischen diesen Tabellen bilden die gerichteten Kanten. Für eine Fremdschlüsselbeziehung von  $R_1$  zu  $R_2$  wird  $R_1$  als Elterntabelle oder Vorgänger und  $R_2$  als Kindtabelle oder Nachfolger bezeichnet. Im Folgenden werden zyklenfreie Schemagraphen angenommen. Zyklen sind dahingehend problematisch, als dass das Hinzufügen eines einzelnen Tupels zu unvorhersehbar großen Synopsen führen kann. Diese Einschränkung ist jedoch wenig restriktiv, da Zyklen in Data-Warehouse-Szenarien unüblich sind. Die relationale Abbildung multidimensionaler Datenwürfel – dem logischen Konzept analytischer Datenbanken – erfolgt üblicherweise in Form von Star- oder Snowflake-Schemata. Dabei werden die zu analysierenden Kennzahlen in einer (großen) Faktentabelle gespeichert, die beschreibenden Informationen befinden sich in einer Vielzahl von (kleineren) Dimensionstabellen. Zwischen der Faktentabelle und den Dimensionstabellen – bei Snowflake-Schemata auch zwischen Dimensionstabellen einer Dimension – bestehen Fremdschlüsselbeziehungen; Szenarien mit mehreren Faktentabellen, welche über Dimensionstabellen miteinander in Beziehung stehen, werden als *Fact Constellation* oder Galaxy-Schema bezeichnet. [8, 11]

## 2.2 Idee und prinzipieller Ablauf

Linked-Bernoulli-Synopsen (LBS) stellen schemaweite Synopsen dar, bei der für jede Tabelle einer Datenbank eine gleichverteilte Zufallsstichprobe existiert. Zusätzlich wird innerhalb der Synopse die referenzielle Integrität garantiert. Dadurch ist es möglich, Anfragen mit beliebigen Fremdschlüsselverbunden näherungsweise zu beantworten. Dies wird durch folgenden Synopsenaufbau erreicht: Eine Synopse  $\Psi$  hält für jede Tabelle des Schemas eine Tabellensynopse. Jede Tabellensynopse  $\Psi_R$  einer Tabelle  $R$  besteht dabei aus zwei Elementen:

- einer gleichverteilten Zufallsstichprobe  $SP_R$ . Diese Stichprobe wird verwendet, um Anfragen (mit und ohne Fremdschlüsselverbundoperationen) an diese Tabelle näherungsweise zu beantworten.
- einer Referenztablette  $RT_R$ . Die Referenztabellen dienen der Erhaltung der referenziellen Integrität. Sie beinhalten all die Tupel, die (1) durch eine beliebige andere Tabellensynopse referenziert werden und (2) noch nicht in die Stichprobe aufgenommen wurden.

Da die Tupel der Referenztablette nicht direkt für die Anfrageverarbeitung verwendet werden können, können sie als Mehraufwand betrachtet werden. Linked-Bernoulli-Synopsen minimieren diesen Mehraufwand basierend auf folgender Beobachtung: Jedes Tupel, welches referenziert wird, muss in jedem Fall gespeichert werden, entweder in der Stichprobe oder aber in der Referenztablette. Da ein Tupel in der Stichprobe für die Anfrageverarbeitung von höherem Wert ist, werden die Stichproben bei LBS in Richtung referenzierter Tupel verzerrt. Im Falle von Speicherplatzbeschränkten Synopsen führt die Minimierung des Mehraufwandes dabei zu größeren Stichproben, welche wiederum in kleineren Schätzfehlern resultieren.

An dieser Stelle folgt ein Überblick über den Prozess der Stichprobenerhebung von LBS; weitere Details des Algorithmus sind in [14] zu finden. Sei  $q_R$  die Stichprobenrate der Tabelle  $R$ ; um die Uniformität der Stichprobe zu erhalten, muss jedes Tupel  $t \in R$  mit Wahrscheinlichkeit  $q_R$  in  $SP_R$  aufgenommen werden. Weiterhin wird jedes Tupel  $t \in R$  mit einer Wahrscheinlichkeit  $pRef_R(t)$  referenziert. Der Wert von  $pRef_R(t)$  ergibt sich aus den Stichprobenraten und den Referenzierungswahrscheinlichkeiten der referenzierenden Tupel; Tupel die nicht referenziert werden, haben eine Referenzierungswahrscheinlichkeit von  $pRef_R(t) = 0$ .

Der Stichprobenerhebungsprozess erfolgt entlang des Schemagraphen von oben nach unten. Damit sind jeweils alle Vorgänger der aktuell bearbeiteten Tabelle bereits abgearbeitet. Die Aufnahme eines Tupels in die Synopse hängt von (1) seiner Referenzierungswahrscheinlichkeit und (2) dem Fakt, ob das Tupel bereits durch einen Vorgänger referenziert wird, ab. Für jedes Tupel  $t$  ergeben sich drei mögliche Fälle:

1. Fall: Die Referenzierungswahrscheinlichkeit des Tupels entspricht der Stichprobenrate ( $pRef_R(t) = q_R$ ). In diesem Fall wird das Tupel genau dann in die Stichprobe eingefügt, wenn es referenziert wird; andernfalls wird es ignoriert. Offensichtlich ist die Aufnahmewahrscheinlichkeit in die Stichprobe damit gleich  $q_R$ .
2. Fall: Die Referenzierungswahrscheinlichkeit des Tupels ist kleiner als die Stichprobenrate ( $pRef_R(t) < q_R$ ). Wird  $t$  referenziert, so wird es in die Stichprobe aufgenommen. Wird  $t$  nicht referenziert, wird es – damit die Aufnahmewahrscheinlichkeit letztendlich wieder  $q_R$  entspricht – mit der Wahrscheinlichkeit  $(q_R - pRef_R(t)) / (1 - pRef_R(t))$  in die Stichprobe aufgenommen; andernfalls wird es ignoriert.
3. Fall: Die Referenzierungswahrscheinlichkeit des Tupels ist größer als die Stichprobenrate ( $pRef_R(t) > q_R$ ). Wird  $t$  nicht referenziert, kann es bedenkenlos ignoriert werden. Wird  $t$  referenziert, so wird es mit Wahrscheinlichkeit  $q_R / pRef_R(t)$  in die Stichprobe

und andernfalls in die Referenztabelle aufgenommen. Somit ist auch in diesem Fall die Aufnahmewahrscheinlichkeit des Tupels in die Stichprobe gleich  $q_R$ . Dies ist damit der *einzig*e Fall, in dem ein Tupel in die Referenztabelle aufgenommen werden kann.

*Beispiel 1.* Gegeben sei das Beispielschema aus Abb. 1 mit den drei Tabellen  $A$ ,  $B$  und  $C$ . Fremdschlüsselbeziehungen bestehen zwischen  $A.FK$  und  $B.PK$  sowie zwischen  $B.FK$  und  $C.PK$ . Gesucht sei eine Linked-Bernoulli-Synopse für  $q_A = q_B = q_C = 0,5$ . Da Relation  $A$  keine Elterntabelle besitzt, gilt  $pRef_A(t) = 0, \forall t \in A$ . Somit ergibt sich stets der 2. Fall und da die Tupel nicht referenziert werden, werden sie mit der Wahrscheinlichkeit  $(q_A - pRef_A(t)) / (1 - pRef_A(t)) = q_A$  in die Stichprobe  $SP_A$  aufgenommen. Anschließend können die Tupel von Tabelle  $B$  betrachtet werden: Das Tupel  $b_1$  wird von Tupel  $a_1 \in A$  referenziert und somit ist die Referenzierungswahrscheinlichkeit  $pRef_B(b_1)$  gleich der Aufnahmewahrscheinlichkeit von  $a_1 = q_A = q_B$ . Dies entspricht dem 1. Fall und  $b_1$  wird genau dann in  $SP_B$  aufgenommen, wenn auch  $a_1$  in  $SP_A$  aufgenommen wurde. Tupel  $b_2$  wird von zwei Tupeln referenziert, die Referenzierungswahrscheinlichkeit ist somit gleich der Wahrscheinlichkeit, dass mindestens eines der Tupel in die Stichprobe aufgenommen wurde, also  $pRef_B(b_2) = 1 - (1 - q_A)^2 = 0,75$ . Dies entspricht dem 3. Fall. Wird  $b_2$  nicht referenziert, d. h. weder  $a_2$  noch  $a_3$  wurden in  $SP_A$  aufgenommen, wird  $b_2$  ignoriert; andernfalls wird  $b_2$  mit der Wahrscheinlichkeit  $q_B / pRef_B(b_2) = 0,67$  in die Stichprobe und mit der Wahrscheinlichkeit von  $1 - q_B / pRef_B(b_2) = 0,33$  in die Referenztabelle aufgenommen. Für die verbleibenden Tupel aus  $B$  wird analog vorgegangen. Abschließend wird dieser Prozess für die Tupel aus Tabelle  $C$  entsprechend durchgeführt.

Die Idee von LBS kann wie folgt zusammengefasst werden: Im Stichprobenerhebungsprozess wird Zufall in die resultierenden Stichproben eingeführt. Dieser Zufall wird von LBS bei der Stichprobenerhebung der entsprechenden

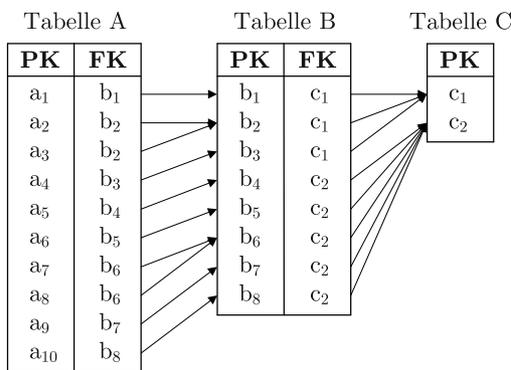


Abb. 1 Beispielschema mit  $V = \{A, B, C\}$  und  $E = \{(A, B), (B, C)\}$

Kindtabellen wiederverwendet. Dadurch wird die Größe der Referenztabelle – und somit des Mehraufwandes – minimiert. Es ist zu beachten, dass durch die Verwendung von LBS Korrelationen zwischen den Stichproben fremdschlüsselverbundener Tabellen eingeführt werden. Nichtsdestotrotz sind die resultierenden Stichproben der einzelnen Tabellen gleichverteilte Zufallsstichproben.

### 3 Allokationsschemata

Mit dem Verfahren zur Erstellung von Linked-Bernoulli-Synopsen werden über die einzelnen Tabellen eines Schemas hinweg statistisch korrekte gleichverteilte Zufallsstichproben erzeugt. Dabei wurde bisher angenommen, dass die gewünschten Stichprobenraten der einzelnen Tabellen vorgegeben sind. Im Allgemeinen wird ein Synopsen-Designer jedoch nicht die einzelnen Stichprobenraten spezifizieren, sondern üblicherweise eine Speicherplatzbeschränkung für die gesamte schemaweite Synopse angeben wollen.<sup>1</sup> Folglich muss der verfügbare Speicherplatz zwischen den Tabellensynopsen aufgeteilt werden – ein herausfordernder Prozess, da die Änderung *einer* Synopsengröße die Größe *vieler* anderer Tabellensynopsen beeinflussen kann.

Bei der Aufteilung des verfügbaren Speicherplatzes auf die Tabellensynopsen können verschiedene Parameter als Einflussgrößen betrachtet werden. Direkt aus den Basisdaten sind dies beispielsweise die Kardinalitäten der Tabellen sowie die Verteilung der Fremdschlüsselattributwerte. Des Weiteren kann – sofern vorhanden – Workload-Information für den Allokationsprozess herangezogen werden. Als mögliche Zielgrößen bei der Allokation lassen sich die einzelnen Stichprobenraten, die Stichprobengrößen sowie der resultierende Mehraufwand spezifizieren.

Im Folgenden werden verschiedene Möglichkeiten vorgestellt, basierend auf den genannten Eingangsparametern die verschiedenen Zielparameter zu berechnen. Dabei wird von folgender Notation Gebrauch gemacht: Bestehe ein Datenbankenschema aus den Tabellen  $R_1, \dots, R_n$  und seien  $q_1, \dots, q_n$  die entsprechenden Stichprobenraten, dann bezeichnet  $q = (q_1, \dots, q_n)$  einen Vektor dieser Stichprobenraten. Sei weiterhin die Speicherplatzbeschränkung durch die Funktion  $g(q) \leq M$  gegeben. Um die nachfolgende Diskussion zu vereinfachen, wird die Speicherplatzbeschränkung in Anzahl zu speichernder Tupel angenommen. Folglich ist es das Ziel, für eine gegebene Speicherplatzbeschränkung  $M$  die Stichprobenraten  $q$  so zu bestimmen, dass mit hoher Wahrscheinlichkeit  $|\Psi| \leq M$  gilt.<sup>2</sup>

<sup>1</sup> Ein analoges Vorgehen wird im Umfeld materialisierter Sichten ebenfalls verfolgt.

<sup>2</sup> Durch das Bernoulli-Vorgehen kann eine Speicherplatzbeschränkung nicht garantiert werden.

Bei den im Folgenden vorgestellten Allokationsschemata gilt folgender Zusammenhang zwischen den Stichprobenraten (bzw. -größen) und dem Mehraufwand:

- Für gegebene Stichprobenraten ist der Mehraufwand minimal; dies ist eine Folge der Verwendung des LBS-Algorithmus.
- Das Verhältnis der einzelnen Stichprobenraten hat maßgeblichen Einfluss auf den Mehraufwand. So können diese in der Art angepasst werden, dass der (wiederum für diese Stichprobenraten minimale) Mehraufwand minimiert wird.

Für die präsentierten Allokationsschemata wird jeweils eine Beispielinstantz einer Linked-Bernoulli-Synopse der Größe  $|\Psi| = 10$  Tupel für das Beispielschema aus Abb. 1 angegeben. Dabei wird im oberen Teil der Abbildung die Stichprobe und im unteren Teil die Referenztable der jeweiligen Tabellensynopse dargestellt.

### 3.1 Workload-basierte Allokation

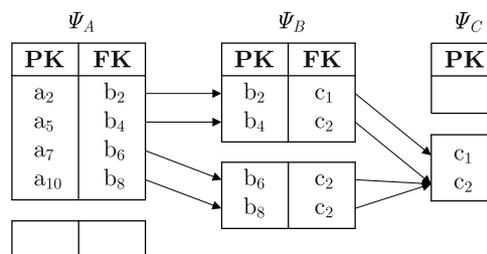
Steht Workload-Information zur Verfügung, kann dieses Anwendungswissen in den Allokationsprozess einfließen. Basierend auf den Auftrittshäufigkeiten in den Anfragen werden den einzelnen Tabellen Gewichte  $w_i$  zugewiesen. Je nach Zielparame-ter werden anschließend die Stichprobenraten  $q_i$  oder die Stichprobengrößen  $n_i$  in der Art gewählt, dass sie das Verhältnis dieser Gewichte widerspiegeln. Eine Kopplung der Größe des Mehraufwandes an die erwartete Anfragemlast scheint wenig sinnvoll.

Seien  $w_1, \dots, w_n$  die Gewichte der entsprechenden Tabellen und gelte weiterhin, dass die Gewichte normiert sind, d. h. sie summieren sich zu 1 auf. Unter Vernachlässigung des durch die Referenztabellen entstehenden Mehraufwandes können die Stichprobenraten mittels

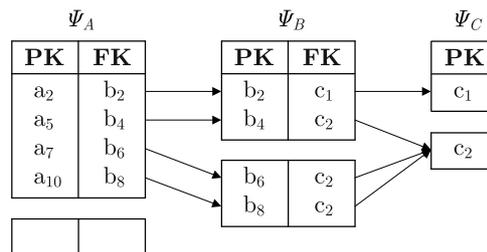
$$q_i = w_i \frac{M}{|R_i|}$$

bestimmt werden. Die Berechnung der Stichprobengrößen entsprechend der Workload-Gewichte erfolgt analog mittels  $n_i = w_i \cdot M$ . Ergibt diese Berechnung  $q_i > 1$  oder  $n_i > |R_i|$ , wird  $q_i = 1$  bzw.  $n_i = |R_i|$  gesetzt und die Berechnung für die verbleibenden Stichprobenraten bzw. -größen wiederholt.

Da bisher der Mehraufwand ignoriert wurde, müssen zusätzlich alle Stichprobenraten bzw. -größen mit einem konstanten Faktor  $c$  herunterskaliert werden, um die Speicherplatzbeschränkung nicht zu überschreiten. Dafür wird  $c$  mittels binärer Suche so lange angepasst, bis mit hoher Wahrscheinlichkeit  $g(q) \leq M$  gilt. Allerdings ist dieses Vorgehen teuer: Die Auswertung der Begrenzungsfunktion  $g$  zieht das Lesen aller Tabellen des Schemas nach sich. Daher ist eine geeignete Abschätzung des Mehraufwandes für die



**Abb. 2** Allokation mit Workload-basierten Stichprobenraten ( $w_A = 0,6$ ,  $w_B = 0,3$ ,  $w_C = 0,1$ ):  $q_A = 42,4\%$ ,  $q_B = 21,2\%$ ,  $q_C = 7,1\%$ ; Mehraufwand: 4 Tupel



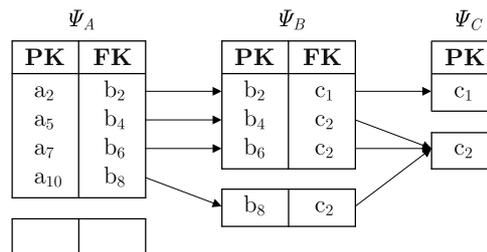
**Abb. 3** Allokation mit Workload-basierten Stichprobengrößen ( $w_A = 0,6$ ,  $w_B = 0,3$ ,  $w_C = 0,1$ ):  $q_A = 42,6\%$ ,  $q_B = 26,7\%$ ,  $q_C = 35,5\%$ ; Mehraufwand: 3 Tupel

jeweiligen Stichprobenraten sinnvoll. Methoden dafür werden in Abschn. 4 betrachtet.

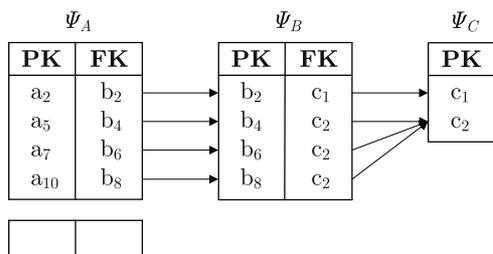
Abbildung 2 zeigt eine Beispielsynopse für die Allokation mit Workload-basierten Stichprobenraten; eine Beispielsynopse für die Allokation mit Workload-basierten Stichprobengrößen ist in Abb. 3 dargestellt.

### 3.2 Kardinalitätenbasierte Allokation

Ist keine Workload-Information verfügbar, sind andere Ansätze für die Bestimmung der Zielparame-ter erforderlich. Bei der wahrscheinlich intuitivsten Art der Speicherplatzaufteilung wird jeder Tabelle die gleiche Stichprobenrate zugewiesen; die Stichprobengrößen sind damit proportional zu den Tabellengrößen. Hierfür werden alle  $q_i$  durch eine Konstante  $c$  spezifiziert und es ist erneut das Ziel,  $c$  so zu bestimmen, dass  $g(q) \leq M$  gilt. Eine entsprechende Beispielsynopse ist in Abb. 4 zu sehen.



**Abb. 4** Allokation mit gleichen Stichprobenraten:  $q_A = 42,3\%$ ,  $q_B = 42,3\%$ ,  $q_C = 42,3\%$ ; Mehraufwand: 2 Tupel



**Abb. 5** Allokation mit gleichen Stichprobengrößen:  $q_A = 38,5\%$ ,  $q_B = 48,2\%$ ,  $q_C = 100\%$ ; kein Mehraufwand

Analog dazu lässt sich auch eine Allokation mit konstanten Stichprobengrößen erzeugen. Die Stichprobenraten sind somit umgekehrt proportional zu den Kardinalitäten der Basistabellen. Eine Beispielsynopse für dieses Allokationschema zeigt Abb. 5.

Eine sinnvolle Kopplung der Höhe des Mehraufwandes an die Kardinalitäten scheint auch hier nicht gegeben.

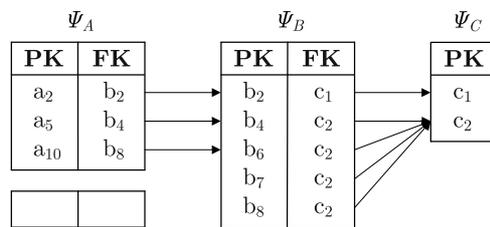
### 3.3 Verteilungsbasierte Allokation

Die kardinalitätenbasierte Allokation führt zu unvorhersehbar großem Mehraufwand. Ein intelligenteres Vorgehen fokussiert daher direkt den Mehraufwand. Hierfür werden drei mögliche Ansätze vorgeschlagen.

*Minimierung des Mehraufwandes* Offensichtlich ist es erstrebenswert, den Mehraufwand einer Synopse zu minimieren. Dies kann (indirekt) durch die Maximierung der Stichprobenraten erreicht werden. Formuliert als Optimierungsproblem kann der geometrische Mittelwert der Stichprobenraten als Bewertungsfunktion herangezogen werden:

$$f_{\text{GEO}}(q_1, \dots, q_n) = \sqrt[n]{q_1 q_2 \dots q_n}.$$

Die Begrenzungsfunktion ist gegeben durch  $g(q)$ . Unglücklicherweise sind sowohl  $f_{\text{GEO}}$  als auch  $g$  nichtlineare Funktionen, wodurch auch das Optimierungsproblem nichtlinear ist. Die Monotonie beider Funktionen vereinfacht jedoch das Problem: Mithilfe des Polyblock-Algorithmus [23] lassen sich Lösungen monotoner Optimierungsprobleme approximieren. Allerdings ist die Zeitkomplexität des Algorithmus exponentiell bezüglich der Anzahl der Tabellen, wodurch dieser bei steigender Anzahl von Tabellen schnell zu rechenintensiv wird. Und selbst bei einer kleinen Anzahl von Tabellen ist der Algorithmus durch die sehr häufig erforderliche Auswertung der Begrenzungsfunktion  $g$  teuer. Erneut ist eine gute Abschätzung der Synopsegröße für gegebene Stichprobenraten erforderlich, um diesen Algorithmus bezüglich des Berechnungsaufwandes anwendbar zu machen. In Abb. 6 ist eine Beispielsynopse mit minimiertem Mehraufwand dargestellt.



**Abb. 6** Allokation mit minimiertem Mehraufwand:  $q_A = 34,4\%$ ,  $q_B = 57,0\%$ ,  $q_C = 100\%$ ; kein Mehraufwand

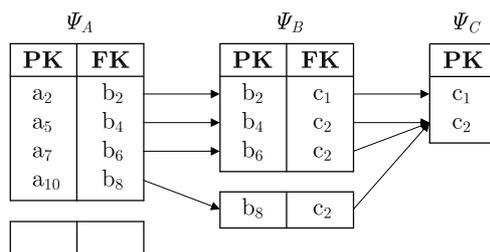
*Beschränkung des Mehraufwandes* Zusätzlich zu dieser parameterfreien Allokationsstrategie kann ein Synopsen-Designer auch an der direkten Beschränkung des Mehraufwandes interessiert sein. Wie der LBS-Algorithmus zeigt, wird der Mehraufwand einer Tabelle durch das Verhältnis der Stichprobenrate der Tabelle und der Referenzierungswahrscheinlichkeiten der einzelnen Tupel bestimmt. Sei  $r$  die relative Grenze für den Mehraufwand bezogen auf die Stichprobengröße. Basierend auf den Referenzierungswahrscheinlichkeiten der Tupel aus Tabelle  $R_i$  müssen die  $q_i$  so gewählt werden, dass mit hoher Wahrscheinlichkeit  $|RT_i| \leq r|SP_i|$  gilt. Aus dem LBS-Algorithmus folgt, dass ein Tupel  $t \in R_i$  nur dann in die Referenztable aufgenommen werden kann, wenn  $\text{pRef}_i(t) > q_i$  (3. Fall). In diesem Fall ist die Wahrscheinlichkeit, dass  $t$  in die Referenztable eingefügt wird:

$$\text{pRefTab}_i(t) = \text{pRef}_i(t) \left( 1 - \frac{q_i}{\text{pRef}_i(t)} \right). \quad (1)$$

Folglich lässt sich für gegebene Referenzierungswahrscheinlichkeiten der Erwartungswert für den Mehraufwand für verschiedene  $q_i$  wie folgt berechnen:

$$E(|RT_i|) = \sum_{j=1}^{|R_i|} \text{pRefTab}_i(t_j). \quad (2)$$

Während eines initialen Lesens der Daten werden die Referenzierungswahrscheinlichkeiten gesammelt. Darauf aufbauend wird das Verhältnis der Stichprobenraten in der Art bestimmt, dass der Mehraufwand die vorgegebene Grenze



**Abb. 7** Allokation mit beschränktem Mehraufwand (max. 33,3% Mehraufwand pro Tabellensynopse):  $q_A = 42,5\%$ ,  $q_B = 36,6\%$ ,  $q_C = 69,7\%$ ; Mehraufwand: 1 Tupel

**Tabelle 1** Allokationsschemata:  
 Zusammenhang der Parameter

	Stichprobenraten $q_i$	Stichprobengrößen $n_i$	Mehraufwand $o$
Workload-Gewichte $w_i$	$q_i \propto w_i$	$n_i \propto w_i$	—
Tabellenkardinalitäten $N_i$	$q_i \propto 1/N_i$	$n_i \propto N_i$	—
Verteilung der Fremdschlüsselattributwerte	$\max(f_{\text{GEO}})$	$\max(f_{\text{GEO}})$	$\max(f_{\text{GEO}}),  o  = r$

einhält. Über eine schrittweise Skalierung der Stichprobenraten wird anschließend  $g(q) \leq M$  erreicht. Abbildung 7 zeigt eine Beispielsynopse mit 10% Mehraufwand.

*Mehraufwandslose Allokation* Die Idee dieses Ansatzes kann dahingehend erweitert werden, dass Referenztabellen vollständig vermieden werden, um somit alle in der Synopse gespeicherten Tupel direkt für die näherungsweise Anfrageverarbeitung verwenden zu können. Dies wird erreicht, indem die Stichprobenraten so gewählt werden, dass für jedes Tupel  $t_i$  jeder Tabelle  $R_j$  gilt  $\text{pRef}_j(t_i) \leq q_j$ . Es ist zu beachten, dass dieses Vorgehen strenger als das oben beschriebene mit einer entsprechenden Mehraufwandsbegrenzung von  $r = 0$  ist: Während letzteres Referenztabellen mit hoher Wahrscheinlichkeit vermeidet, garantiert dieses Vorgehen  $|RT_i| = 0, \forall i$ . Die Berechnung der Allokation ist ähnlich, nur wird die Stichprobenrate einer Tabelle hier auf die maximale Referenzierungswahrscheinlichkeit aller Tupel dieser Tabelle gesetzt. Für das Beispielszenario resultiert die mehraufwandslose Allokation in der gleichen Speicherplatzaufteilung wie die Allokation mit minimiertem Aufwand; als Beispielsynopse kann daher die Synopse aus Abb. 6 betrachtet werden; die resultierenden Stichprobenraten sind ähnlich ( $q_A = 35,5\%$ ,  $q_B = 57,2\%$ ,  $q_C = 98,6\%$ ). Tabelle 1 fasst noch einmal den Zusammenhang zwischen den Eingangs- und den Zielparametern der vorgestellten Allokationsschemata zusammen. Dabei geben die Zeilen die verschiedenen Eingangsparameter und die Spalten die betrachteten Zielparameter an. In den einzelnen Zellen ist jeweils zu sehen, wie die Eingangsparameter die Zielparameter bestimmen.

#### 4 Abschätzen der erwarteten Synopsengröße

Im vorangegangenen Abschnitt wurden verschiedene Allokationsschemata vorgestellt. Sie alle benötigen für die Bestimmung der einzelnen Stichprobenraten die resultierende Synopsengröße und damit die Größe des erwarteten Mehraufwandes. Die exakte Berechnung dieser Größen für gegebene Stichprobenraten  $q$  erfordert jedoch das Lesen aller Tabellen. Während dies für einige Verfahren akzeptabel sein kann, da die Anzahl der Iterationsschritte für die Berechnung der Synopsengröße klein ist, wird spätestens für den Polyblock-Algorithmus der Aufwand inakzeptabel groß. Andererseits ist gerade die

Minimierung des Mehraufwandes – wofür der Polyblock-Algorithmus benötigt wird – ein hervorragendes Allokationsschema, wenn Workload-Information fehlt und der Synopsen-Designer keine zusätzlichen Parameter vorgeben möchte: Es maximiert ohne jegliches Anwendungswissen die Größe der Stichproben (und minimiert somit den Schätzfehler) basierend auf der Charakteristik der Daten.

*Beispiel 2.* Für die Berechnung der exakten erwarteten Synopsengröße einer Linked-Bernoulli-Synopse des Beispielschemas aus Abb. 1 mit  $q_A = q_B = q_C = 0,5$  ist das Lesen aller drei Tabellen  $A$ ,  $B$  und  $C$  erforderlich; das Ergebnis beträgt  $g(q_A, q_B, q_C) = 11,42$ ; für Details siehe [14].

##### 4.1 Vernachlässigung des Mehraufwandes

Bereits in [14] wird die Abschätzung von  $g$  durch die einfache Vernachlässigung der Referenztabellen betrachtet. Diese Abschätzung stellt eine Untergrenze der Synopsengröße dar. Für gegebene Stichprobenraten  $q_1, \dots, q_n$  kann diese Untergrenze  $g^-$  wie folgt berechnet werden:

$$g^-(q_1, \dots, q_n) = |R_1|q_1 + \dots + |R_n|q_n \leq g(q_1, \dots, q_n).$$

Diese Abschätzung trägt zu einer signifikanten Beschleunigung der Bestimmung der erwarteten Synopsengröße bei, da  $g^-$  sehr schnell und ohne Zugriff auf die Datenbank berechnet werden kann. Allerdings können durch die Vernachlässigung der Referenztabellen die resultierenden Synopsen die Speicherplatzbeschränkung überschreiten. In [14] wird ebenfalls gezeigt, dass die Größe der Referenztabellen für die Allokation mit gleichen Stichprobengrößen häufig vernachlässigbar ist; dies gilt offensichtlich auch für die Allokation mit minimiertem Mehraufwand. Für andere Allokationsschemata kann diese Unterschätzung jedoch signifikant sein.

*Beispiel 3.* Die Abschätzung der erwarteten Synopsengröße unter Vernachlässigung des Mehraufwandes ergibt  $g^-(q_A, q_B, q_C) = |A|q_A + |B|q_B + |B|q_B = 10$ .

##### 4.2 Annahme gleichverteilter Referenzen

Eine erweiterte Form der Abschätzung der Synopsengröße zieht die Referenztabellen mit in Betracht. Die zu Grunde liegende Idee ist dabei die Annahme der Gleichverteilung der Fremdschlüsselattributwerte. Sei  $fk_{i \rightarrow j}$  die Menge der

Fremdschlüsselattributwerte von  $R_i$ , welche Tupel in  $R_j$  referenzieren. Durch die Gleichverteilungsannahme hat jeder Wert in  $fk_{i \rightarrow j}$  die gleiche Auftrittshäufigkeit in  $R_i$ , dargestellt durch  $n_{i \rightarrow j}$ . Somit haben alle Tupel  $t \in R_j$  die gleiche Referenzierungswahrscheinlichkeit:

$$pRef_j(t) = 1 - (1 - pSel_i(t))^{n_{i \rightarrow j}}.$$

Der Wert  $pSel_i(t)$  gibt die Aufnahmewahrscheinlichkeit des Tupels  $t$  in die Tabellensynopse  $\Psi_i$  an und wird durch  $pSel_i(t) = \max\{q_i, pRef_i(t)\}$  berechnet, für mehr Details sei auf [14] verwiesen. Da  $pSel_i(t)$  nicht ohne das vollständige Lesen der Tabellen berechnet werden kann – was dem Aufwand einer exakten Berechnung entspräche –, wird im Weiteren  $q_i$  anstelle von  $pSel_i(t)$  verwendet. Folglich wird die Referenzierungswahrscheinlichkeit  $pRef_{R_j}(t)$  durch

$$p\tilde{Ref}_j(t) = 1 - (1 - q_i)^{n_{i \rightarrow j}}$$

angenähert. Anschließend bietet Gl. 1 eine Näherung der Aufnahmewahrscheinlichkeit  $pRefTab_j(t)$  von  $t$  in die Referenztablette und liefert zusammen mit der Anzahl referenzierter Tupel  $|fk_{i \rightarrow j}|$  die erwartete Größe der Referenztablette von  $\Psi_j$ :

$$E(|RT_j|) = |fk_{i \rightarrow j}| \cdot pRefTab_j(t).$$

Dies ist eine Vereinfachung von Gl. 2 für gleiche  $pRefTab_{R_j}(t)$  unter Beachtung, dass es auch unreferenzierte Tupel in  $R_j$  geben kann. Angewendet auf alle Tabellen der Datenbank ergibt

$$g^-(q_1, \dots, q_n) = \sum_{i=1}^n (|R_i|q_i + E(|RT_i|))$$

eine Schätzung der Synopsengröße  $g$ .

*Beispiel 4.* Für die Abschätzung unter der Annahme gleichverteilter Referenzen ergibt sich für  $|RT_A| = 0$ , da  $A$  keine Elterntabelle besitzt. Die durchschnittliche Auftrittshäufigkeit der Fremdschlüsselattributwerte in  $A$  beträgt  $n_{A \rightarrow B} = 10/8 = 1,25$ . Damit ergibt sich für die durchschnittliche Referenzierungswahrscheinlichkeit der Tupel  $t \in B$ :  $p\tilde{Ref}_B(t) = 1 - (1 - q_B)^{n_{A \rightarrow B}} = 0,58$ , und für den erwarteten Mehraufwand von  $B$  folgt:  $E(|RT_B|) = 0,64$ . Für den Mehraufwand von  $C$  gilt entsprechend:  $n_{B \rightarrow C} = 4$ ,  $p\tilde{Ref}_C(t) = 0,94$  und damit  $E(|RT_C|) = 0,88$ . Für die erwartete Synopsengröße folgt damit:  $g^-(q_A, q_B, q_C) = 11,52$ .

#### 4.3 Erweiterte Referenzierungsinformationen

Offensichtlich wird die Schätzung mittels  $g^-$  umso ungenauer, je stärker die Verteilung der Fremdschlüsselattributwerte verzerrt ist. Um dies zu vermeiden, speichert der folgende Ansatz während eines initialen Lesens

für jede Tabelle die Auftrittshäufigkeiten  $h_{v_i}$  der Fremdschlüsselattributwerte  $v_i$ . Für die Abschätzung des Mehraufwandes einer Tabellensynopse  $\Psi_j$  einer Tabelle  $R_j$  mit einer Elterntabelle  $R_i$  wird über alle  $t \in R_j$  iteriert und die Referenzierungswahrscheinlichkeit

$$p\tilde{Ref}_j(t) = 1 - (1 - q_i)^{h_{v_i}}$$

mit den entsprechenden  $h_{v_i}$  aus  $R_i$  angenähert. Ebenso wie im vorigen Ansatz wird  $q_i$  anstelle von  $pSel_i(t)$  verwendet und der erwartete Mehraufwand kann mit Hilfe der Gl. 1 und 2 berechnet werden. Abschließend wird  $g$  mittels

$$g^+(q_1, \dots, q_n) = \sum_{i=1}^n (|R_i|q_i + E(|RT_i|))$$

abgeschätzt. Eine vereinfachte Variante dieses Vorgehens schätzt die Auftrittshäufigkeiten  $h_{v_i}$  aus einem Histogramm ab.

*Beispiel 5.* Wie für  $g^-$  gilt auch für  $g^+$ :  $|RT_A| = 0$ . Die Auftrittshäufigkeit  $h_{v_i}$  der Fremdschlüsselattributwerte  $v_i \in \{b_2, b_6\}$  aus Tabelle  $A$  beträgt  $h_{v_i} = 2$ , die der verbleibenden Fremdschlüsselattributwerte ist  $h_{v_i} = 1$ . Für die Referenzierungswahrscheinlichkeit von  $b_2$  und  $b_4$  aus  $B$  folgt damit  $p\tilde{Ref}_B(t) = 1 - (1 - q_B)^{h_{v_i}} = 0,75$  sowie  $p\tilde{Ref}_B(t) = 0,5$  für die verbleibenden Tupel  $t \in B$ . Der erwartete Mehraufwand von  $B$  ist somit  $E(|RT_B|) = 0,5$ . Die Berechnung des erwarteten Mehraufwandes von  $C$  erfolgt analog und ergibt  $E(|RT_C|) = 0,84$ . Für die erwartete Synopsengröße folgt damit:  $g^+(q_A, q_B, q_C) = 11,34$ .

## 5 Einbettung in Data-Warehouse-Umgebungen

Während sich die letzten Abschnitte mit dem Layout von Linked-Bernoulli-Synopsen beschäftigten, fokussiert dieser Abschnitt deren Einsatz in Data-Warehouse-Umgebungen. Hierfür werden die Wartung der Synopsen beim Laden neuer Daten ins Data-Warehouse sowie eine zusätzliche Optimierung für Aggregationsanfragen betrachtet.

### 5.1 Einfügen neuer Daten

In Data-Warehouse-Szenarien bestehen Änderungen an den Daten üblicherweise aus dem Einfügen neuer Daten in die Faktentabelle. Um nach dem Einfügen einer Vielzahl neuer Daten noch gute näherungsweise Anfrageergebnisse zu erhalten, müssen die Synopsen aktualisiert werden. Da hierbei eine Neuberechnung eine häufig inakzeptabel hohe Systembelastung darstellt, wird im Folgenden betrachtet, wie eine schemaweite Synopse inkrementell gewartet werden kann.

Die effizienteste Möglichkeit der Synopsenwartung behält die Stichprobenraten der bestehenden Synopse bei und fügt die neuen Tupel mit der entsprechenden Wahrscheinlichkeit in die Stichprobe der Faktentabelle ein. Die Tabellensynop-

sen der Kindtabellen müssen nur dann angepasst werden, wenn durch das Einfügen neuer Tupel die referenzielle Integrität der schemaweiten Synopse verletzt wird; die jeweiligen Stichproben sind bereits statistisch korrekt. Für neu in die Tabellensynopse eingefügte Tupel sind jeweils folgende Fälle zu betrachten:

1. Es sind bereits Tupel mit den Werten der Fremdschlüsselattribute dieses Tupels in der Tabellensynopse enthalten. Somit werden keine neuen Kindtupel referenziert und das Einfügen dieses Tupels ist beendet.
2. Mindestens ein Wert der Fremdschlüsselattribute ist noch nicht in der Tabellensynopse enthalten. Hier müssen die entsprechenden Tabellensynopsen der Kindtabellen betrachtet werden:
  - i) Das neu referenzierte Tupel befindet sich bereits in der Tabellensynopse. Damit ist die referenzielle Integrität gewährleistet und das Einfügen des Tupels ist beendet.
  - ii) Das neu referenzierte Tupel befindet sich noch nicht in der Tabellensynopse. In diesem Fall muss es zur Wahrung der referenziellen Integrität in die Tabellensynopse aufgenommen werden. Da die Stichproben der Kindtabellen nicht verändert werden können, ohne deren Uniformität zu verletzen, wird das Tupel in die Referenztablette eingefügt. Anschließend wird für dieses Tupel die Fallunterscheidung wiederholt, wodurch rekursiv entlang der Pfade des Schemagraphen die referenzielle Integrität wiederhergestellt wird.

Der Aufwand der Wartung beschränkt sich auf das Lesen der in Fall 2ii) neu einzufügenden Tupel. Dies kann effizient über einen Index erfolgen, da es sich hierbei um Zugriffe über den Primärschlüssel handelt.

*Beispiel 6.* Gegeben sei die Synopse aus Abb. 6, und werde beim Einfügen neuer Tupel in die (Fakten-)Tabelle  $A$  ein Tupel  $(a_{15}, b_2)$  in die Tabellensynopse  $\Psi_A$  aufgenommen. Da der Fremdschlüsselattributwert  $b_2$  bereits in  $\Psi_A$  enthalten ist, sind keine weiteren Änderungen an der schemaweiten Synopse erforderlich. Beim Einfügen eines weiteren Tupels  $(a_{21}, b_7)$  in  $\Psi_A$  wird zunächst festgestellt, dass sich der Fremdschlüsselattributwert  $b_7$  noch nicht in  $\Psi_A$  befindet; die Überprüfung von  $\Psi_B$  ergibt jedoch, dass  $b_7$  bereits in die Synopse aufgenommen wurde und somit keine weiteren Schritte erforderlich sind. Erst beim Einfügen eines weiteren Tupels  $(a_{25}, b_3)$  wird festgestellt, dass  $b_3$  weder in  $\Psi_A$  noch in  $\Psi_B$  enthalten ist und somit in  $RT_B$  aufgenommen werden muss.

Ist die Speicherplatzbeschränkung relativ zur Größe der Basisdaten gewünscht, sorgt dieses Vorgehen für automatisch mitwachsende Synopsen. Für absolute Speicherplatzbeschränkungen kann das Verfahren dahingehend erweitert werden, dass die Stichprobenrate der Faktentabelle entspre-

chend der Speicherplatzüberschreitung verringert wird. Anschließend wird aus der bestehenden Stichprobe eine Teilstichprobe mit der neuen Stichprobenrate gezogen. Dies ist möglich, da keine Referenzen auf die Faktentabelle bestehen und es somit nicht zu Verletzungen der referenziellen Integrität kommen kann. Die Einhaltung der Speicherplatzbeschränkung hat allerdings zur Folge, dass die Stichprobenrate der Faktentabellenstichprobe kontinuierlich sinkt.

Für beide Vorgehen ist zu bemerken, dass eine Aufrechterhaltung der spezifischen Eigenschaften einiger Allokationsschemata nicht gewährleistet wird. Hierfür wäre die Neuberechnung der Stichproben erforderlich.

## 5.2 Zusätzliche Optimierung für Aggregationsanfragen

Neben Fremdschlüsselverbunden sind Aggregationen (insbesondere Summe und Durchschnitt) Hauptbestandteil typischer Data-Warehouse-Anfragen. Bei der näherungsweise Anfrageverarbeitung werden diese Aggregate basierend auf den Daten der Stichproben geschätzt. Hierbei können Ausreißer in der Datenbasis zu großen Schätzfehlern führen [6, 20]. Als Ausreißer werden dabei Werte bezeichnet, die sich signifikant von den anderen Werten unterscheiden und einen maßgeblichen Einfluss auf das Aggregat haben. Als Beispiel diene eine hypothetische Firma mit einem Jahresumsatz von 1 Mrd. Euro und einer einzelnen Bestellung von 100 Mio. Euro; diese eine Bestellung ist als einzelnes Tupel in der Datenbank repräsentiert, trägt jedoch 10% zum Jahresumsatz dieser Firma bei. Das Fehlen solcher Ausreißer in der Stichprobe führt zu Unterschätzungen, das Vorhandensein zu Überschätzungen der Aggregate. Infolgedessen ist eine geeignete Repräsentation von Ausreißern in der Synopse von großer Wichtigkeit.

In [6] zeigen Chaudhuri et al., wie durch eine Extraktion vor der Stichprobenerhebung und eine separate Speicherung der Ausreißer der Schätzfehler für das Aggregat signifikant gesenkt werden kann. Dieser Ansatz wurde in [20] durch die Erweiterung der Ausreißerbehandlung über mehrere Aggregationsattribute für den Einsatz in Data-Warehouse-Umgebungen einsetzbar gemacht.

Eine Erweiterung von Linked-Bernoulli-Synopsen um die Ausreißerbehandlung aus [20] ist vielversprechend, wie am Beispiel einer Schätzung eines Mittelwertes gezeigt werden soll:<sup>3</sup> Sei  $\mu_j$  der Mittelwert eines Attributes  $j$  der Tabelle  $R$ . Bei der näherungsweise Anfrageverarbeitung wird  $\mu_j$  mit dem Mittelwert  $\hat{\mu}_j$  aus den Tupeln der Stichprobe geschätzt. Sei  $\sigma_j^2$  die Varianz des Attributes  $j$  in den Basisdaten und  $n$  die Größe der Stichprobe, dann ergibt

$$\sigma_{\hat{\mu}_j}^2 = \frac{\sigma_j^2}{n} \left( 1 - \frac{n}{|R|} \right) \quad (3)$$

<sup>3</sup> Die Schätzung einer Summe ist ähnlich.

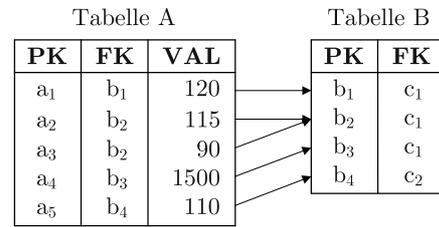
die Varianz des geschätzten Mittelwertes. Dieser Wert spiegelt die Genauigkeit der Schätzung wieder: Je kleiner der Wert, umso genauer ist das approximative Anfrageergebnis. Bei der Erweiterung von Linked-Bernoulli-Synopsen um die Ausreißerbehandlung aus [20] ermöglicht die Minimierung des Mehraufwandes mittels LBS große Stichprobengrößen, die zusätzliche Ausreißerbehandlung sorgt für geringe Varianzen in den den Stichproben zu Grunde liegenden Basisdaten. Wie in Gl. 3 zu sehen ist, haben beide Fakten direkten Einfluss auf die Schätzgenauigkeit. Somit lassen sich mit dem erweiterten Verfahren Data-Warehouse-typische Anfragen mit hoher Genauigkeit approximativ beantworten.

Für die Ausreißerbehandlung in schemaweiten Synopsen wird für jede Tabelle  $R$  die Tabellensynopse  $\Psi_R$  um eine *Ausreißertabelle*  $AT_R$  erweitert. In diesen Ausreißertabellen werden die berechneten Ausreißer vollständig gespeichert. Diese Tupel können direkt für die approximative Anfrageverarbeitung verwendet werden. Zusätzlich muss der LBS-Algorithmus wie folgt leicht angepasst werden, das Vorgehen erfolgt dabei weiterhin von oben nach unten entlang des Schemagraphen: Für die vorgegebenen oder mit Hilfe des Allokationsprozesses bestimmten Stichprobengrößen werden die Ausreißer für die Aggregationsattribute mit dem unveränderten Algorithmus aus [20] berechnet und in den Ausreißertabellen gespeichert. Die Referenzierungswahrscheinlichkeit von durch Ausreißer referenzierten Tupeln  $t \in R$  wird auf  $pRef_t(R) = 1,0$  gesetzt. Anschließend werden die Stichprobenraten auf den verbleibenden Speicherplatz und die verbleibenden Tupel der Basisdaten angepasst: Sei  $O_i$  die Menge der berechneten Ausreißer von  $R_i$ , dann wird die Stichprobenrate von  $R_i$  auf

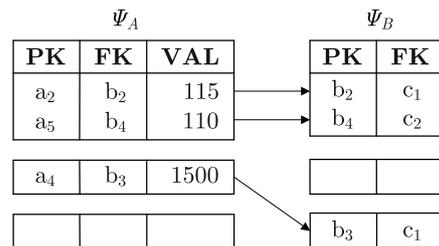
$$q'_i = \frac{q_i |R_i| - |O_i|}{|R_i| - |O_i|}$$

gesetzt. Abschließend werden aus  $R_i \setminus O_i$  die Stichprobe und die Referenztablelle wie bisher berechnet.

*Beispiel 7.* Gegeben sei das angepasste Beispielschema aus Abb. 8 mit den beiden Tabellen  $A$  und  $B$  und einer Fremdschlüsselbeziehung zwischen  $A.FK$  und  $B.PK$  sowie dem Aggregationsattribut  $A.VAL$ . Eine Beispielinstantz einer Linked-Bernoulli-Synopse mit Ausreißerbehandlung für  $A.VAL$  mit  $q_A = 0,6$  und  $q_B = 0,5$  ist in Abb. 9 dargestellt: Die Ausreißerbehandlung identifiziert  $a_4$  als Ausreißer; dieses Tupel wird somit in die Ausreißertabelle  $AT_A$  (in der Mitte der Tabellensynopse dargestellt) eingefügt. Die Stichprobenrate für die verbleibenden Tupel aus  $A$  wird auf  $q'_A = (q_A |R_A| - |O_A|) / (|R_A| - |O_A|) = 0,5$  angepasst, und die Tupel  $a_2$  und  $a_5$  werden in die Stichprobe aufgenommen. Anschließend kann mit Relation  $B$  fortgesetzt werden: Für die Tupel  $b_1$ ,  $b_2$  und  $b_4$  wird analog zu Beispiel 1 vorgegangen; die Wahrscheinlichkeiten sind identisch. Das Tupel  $b_3$  wird mit einer Wahrscheinlichkeit



**Abb. 8** Anpasstes Beispielschema mit einem Aggregationsattribut (VAL) in Tabelle A



**Abb. 9** Linked-Bernoulli-Synopse mit Ausreißerbehandlung; oben: Stichproben, Mitte: Ausreißertabellen, unten: Referenztabellen

von  $pRef_B(b_3) = 1,0$  referenziert und wird somit mit der Wahrscheinlichkeit  $q_B / pRef_B(b_3) = 0,5$  in die Stichprobe und mit der Wahrscheinlichkeit von 0,5 in die Referenztablelle eingefügt.

## 6 Evaluation

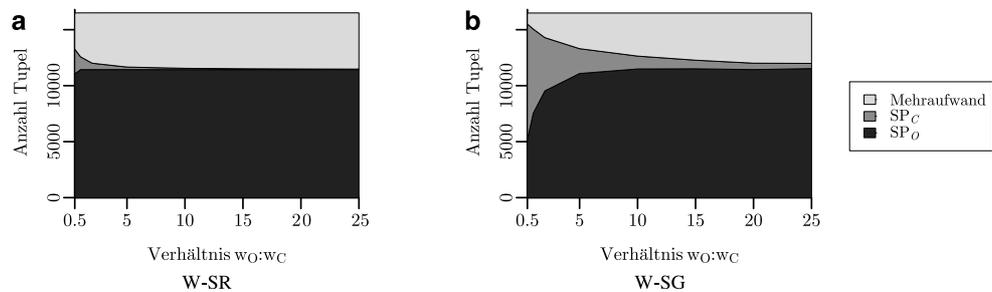
Mit einer Vielzahl von Experimenten wurden die in diesem Artikel vorgestellten Verfahren und Techniken evaluiert. Für die verschiedenen Allokationsschemata wurde deren Einfluss auf die einzelnen Stichprobenraten bzw. -größen und den resultierenden Mehraufwand untersucht; die Genauigkeit und der Aufwand der vorgestellten Verfahren für die Abschätzung der Synopsengröße wurde analysiert, und für die Verwendung von LBS in Data-Warehouse-Umgebungen wurden die Auswirkungen von Einfügeoperationen und der Optimierung für Aggregationsattribute betrachtet.

Die Experimente haben gezeigt, dass die Wahl des Allokationsschemas großen Einfluss auf den Mehraufwand hat und eine zusätzliche Optimierung für Aggregationsattribute die Genauigkeit der Schätzung signifikant verbessern kann.

### 6.1 Experimentierumgebung

Linked-Bernoulli-Synopsen und die in diesem Artikel diskutierten Verfahren wurden in Java 1.6 als Zwischenschicht auf ein Datenbanksystem implementiert. Die Experimente wurden auf einem Linux-Rechner mit zwei Dual Core AMD Opteron Prozessoren mit 2 GHz Taktfrequenz und 9 GB Arbeitsspeicher durchgeführt.

**Abb. 10** Einfluss der Workload-Gewichte auf das Synopsenlayout für eine 1% Synopse (16 500 Tupel)



Als Daten für die Experimente wurden unterschiedlich modifizierte TPC-H-Daten<sup>4</sup> der Größe 1 GB verwendet. Dabei wurden sowohl die Häufigkeitsverteilung der Fremdschlüsselattributwerte als auch die Werteverteilung der Aggregationsattribute so verändert, dass diese einer Zipf-Verteilung folgen. Der  $z$ -Wert der Zipf-Verteilung variierte dabei zwischen den verschiedenen Experimenten.

## 6.2 Allokationsschemata

Der erste Teil der Experimente vergleicht die verschiedenen Allokationsschemata und zeigt deren Auswirkungen auf Mehraufwand und Stichprobenraten bzw. -größen. Hierfür wurden verschiedene Workload-Gewichte, Verteilungen der Fremdschlüsselattribute und Obergrenzen für den Mehraufwand betrachtet.

**Workload-basierte Allokation** Für dieses Allokationsschema wurde der Einfluss der Workload-Gewichte auf das Synopsenlayout untersucht. Dabei wurde für verschiedene Workload-Gewichte jeweils eine 1% Synopse (16 500 Tupel) der Tabellen Orders ( $O$ ) und Customer ( $C$ ) mit einer Schiefe der Fremdschlüsselattributverteilung von  $z = 1$  berechnet. Die Experimente zeigen, dass die jeweiligen Stichproben wie erwartet mit steigenden Workload-Gewichten wachsen. Jedoch wird auch offensichtlich, dass recht schnell der Einfluss der Workload-Gewichte relativ gering wird. Der Grund dafür ist, dass durch die Erhöhung von  $q_O$  die Referenzierungswahrscheinlichkeiten  $p_{\text{Ref}_C}(t)$ ,  $t \in C$  steigen. Damit steigt wiederum die Aufnahmewahrscheinlichkeit in die Referenztabellen und somit auch der Mehraufwand. Folglich kann  $q_O$  nicht weiter ansteigen, ohne zu Überschreitungen der Speicherplatzbeschränkung zu führen.

Für die Kopplung der Stichprobenraten an die Workload-Gewichte (W-SR) zeigt Abb. 10a das Synopsenlayout für verschiedene Verhältnisse der Workload-Gewichte  $w_O$  und  $w_C$ . Bei einem Verhältnis von 0,5 ( $w_O = 0,33$ ,  $w_C = 0,67$ ) beträgt  $q_O = 0,74\%$ ,  $q_C = 1,48\%$  und der Mehraufwand der Synopse 19,59%. Ist das Verhältnis der Workload-Gewichte dagegen 25 ( $w_O = 0,96$ ,  $w_C = 0,04$ ), erhöht sich

die Stichprobenrate  $q_O$  kaum messbar auf 0,76%, während  $q_C$  auf 0,03% fällt und der Mehraufwand auf 30,42% steigt.

Wird als Zielparame-ter die Stichprobengrößen gewählt (W-SG) ist für Verhältnisse der Workload-Gewichte bis zu einem Wert von 5 ein deutlicher Anstieg der Stichprobengröße  $|SP_O|$  zu beobachten; für größere Werte verhält sich der Verlauf ähnlich wie bei W-SR, siehe Abb. 10b. Die Ursache hierfür ist in den anfangs deutlich kleineren Stichprobenraten  $q_O$  zu finden.

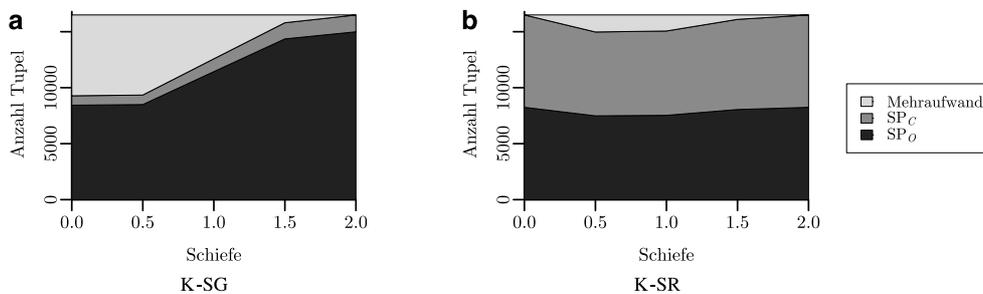
**Kardinalitätenbasierte Allokation** Um den Unterschied der beiden vorgestellten kardinalitätenbasierten Allokationsschemata zu verdeutlichen, wurde im folgenden Experiment der Einfluss der Schiefe der Häufigkeitsverteilung der Fremdschlüsselattributwerte auf das Synopsenlayout untersucht. Erneut wurden 1% Synopsen der Tabellen Orders und Customer berechnet, die Schiefe der Häufigkeitsverteilung variierte zwischen  $z = 0$  (gleichverteilt) und  $z = 2$  (stark verzerrt). Die Ergebnisse sind in Abb. 11 dargestellt. Bestimmen die Tabellenkardinalitäten die Stichprobengrößen (K-SG), d. h. alle Stichprobenraten sind gleich, ist für kleine  $z$ -Werte der Mehraufwand der Synopsen relativ hoch (bis zu 44,31%). Steigt  $z$ , so werden immer mehr Bestellungen von immer weniger Kunden getätigt. Somit fällt statistisch die Anzahl Bestellungen verschiedener Kunden in  $SP_O$ , wodurch weniger Tupel aus C zur Erhaltung der referenziellen Integrität gespeichert werden müssen – der Mehraufwand sinkt und  $SP_O$  und  $SP_C$  können wachsen, siehe Abb. 11a.

Ein deutlich anderes Bild ergibt sich für die Allokation mit gleichen Stichprobengrößen (K-SR). Hierbei ist das Synopsenlayout relativ unabhängig von der Schiefe der Verteilung und der Mehraufwand stets gering (max. 9,24%), siehe Abb. 11b.

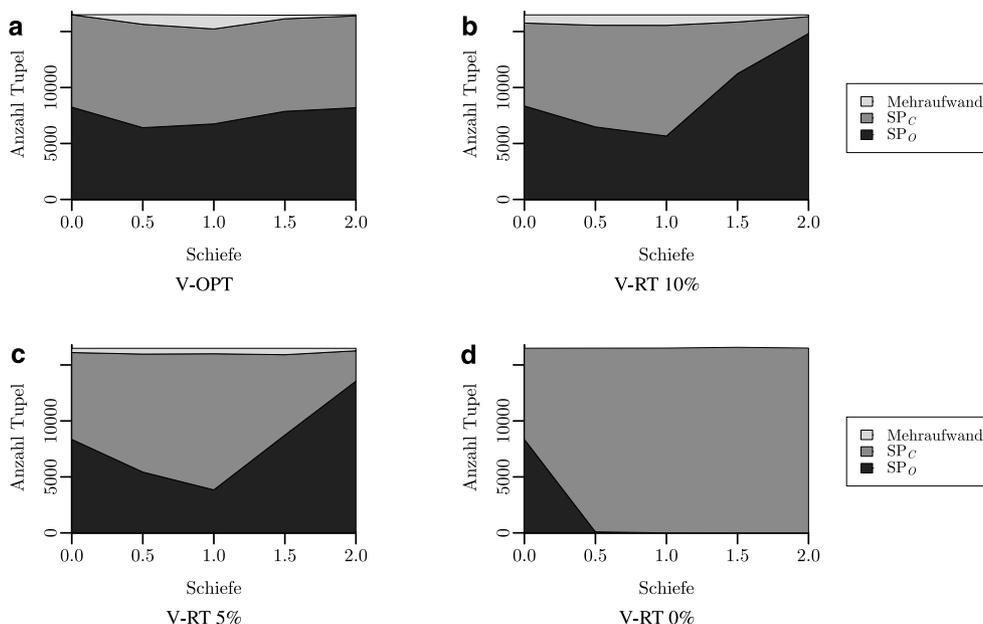
**Verteilungsbasierte Allokation** Bei der verteilungsbasierten Allokation wurde betrachtet, welche Auswirkungen die Mehraufwandsbegrenzung und die Häufigkeitsverteilung der Fremdschlüsselattributwerte auf das Synopsenlayout haben. Wie im vorangegangenen Experiment wurden Synopsen der Größe 1% berechnet und die Schiefe der Verteilung zwischen  $z = 0$  und  $z = 2$  variiert. Der Mehraufwand wurde auf 10% (V-RT 10%) bzw. 5% (V-RT 5%) begrenzt oder

<sup>4</sup> <http://www.tpc.org>

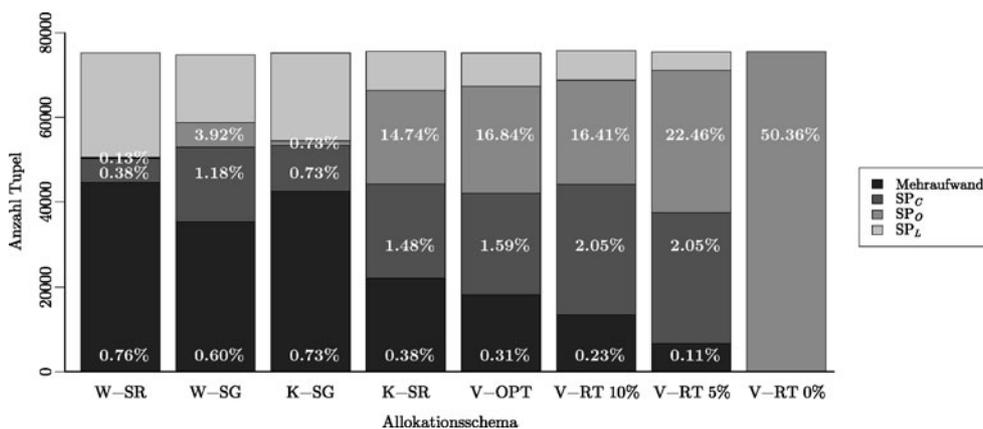
**Abb. 11** Einfluss der Verteilung der Fremdschlüsselattributwerte auf das Synopsenlayout für eine 1% Synopse (16 500 Tupel)



**Abb. 12** Einfluss der Mehraufwandsbegrenzung im Vergleich zur optimalen Allokation



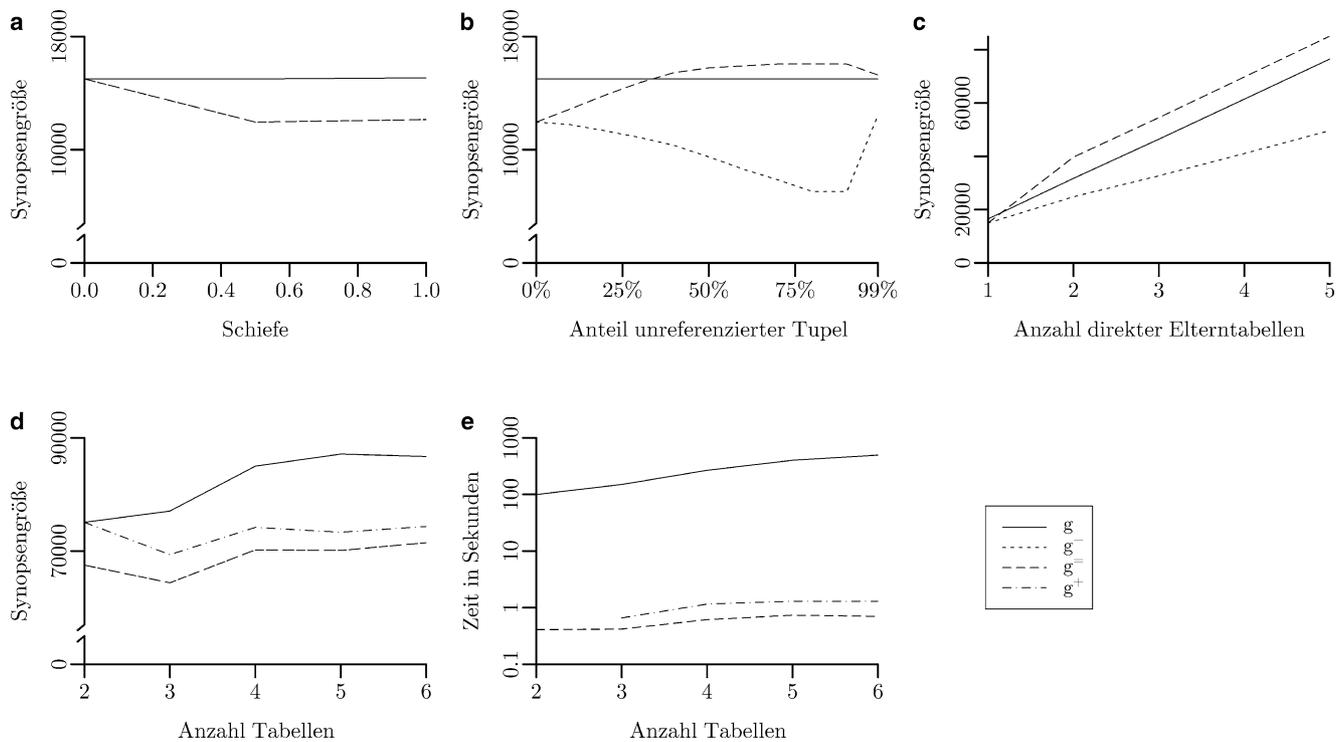
**Abb. 13** Vergleich der Allokationsschemata



ganz vermieden (V-RT 0%). Außerdem wurde die Allokation mit minimiertem Mehraufwand, d. h., die optimale Allokation bezüglich  $f_{GEO}$ , berechnet (V-OPT).

Die Ergebnisse sind in Abb. 12 zusammengefasst. Für V-OPT ist zu erkennen, dass der Mehraufwand für  $z = 1$  am höchsten ist (Abb. 12a). Die Folge davon ist, dass an dieser Stelle bei begrenztem Mehraufwand die Stichprobengröße  $|SP_O|$  am kleinsten und  $|SP_C|$  entsprechend

am größten ist (Abb. 12b und c). Soll der Mehraufwand vollständig vermieden werden, wird die Synopse schnell von  $SP_C$  dominiert, der Anteil von  $SP_O$  wird verschwindend klein (Abb. 12d). Der Grund dafür ist, dass mit steigender Schiefe  $z$ , die maximale Referenzierungswahrscheinlichkeit  $\max(pRef_C(t), t \in C)$  steigt. Um zu garantieren, dass kein Mehraufwand entsteht, muss  $q_C \geq \max(pRef_C(t), t \in C)$  gewählt werden; der Anteil von  $SP_C$  steigt.



**Abb. 14** Vergleich der Abschätzungen der Synopsisgröße

**Zusammenfassung** Abbildung 13 zeigt noch einmal alle Allokationsschemata im Vergleich. Hierfür wurde jeweils eine Synopsis der Größe 1% (76 500 Tupel) für die Tabellen Lineitem ( $L$ ), Orders ( $O$ ) und Customer ( $C$ ) des modifizierten TPC-H-Schemas berechnet. Die Schiefe der Häufigkeitsverteilung der Fremdschlüsselattribute betrug  $z = 1$ , die Workload-Gewichte wurden auf  $w_L = 0,6$ ,  $w_O = 0,3$  und  $w_C = 0,1$  gesetzt. Im direkten Vergleich der Allokationsschemata fällt auf, dass bei W-SR und K-SG<sup>5</sup> der größte Mehraufwand entsteht. Andererseits haben besonders die Allokationsschemata mit geringem Mehraufwand relativ niedrigere Stichprobenraten für die Faktentabelle. Hieraus sind deutlich die Auswirkungen einer Erhöhung der Stichprobenrate der Faktentabelle auf den Mehraufwand zu erkennen. Weiterhin ist zu beobachten, dass die Allokationsschemata K-SR und V-OPT ein recht ähnliches Synopsenlayout zur Folge haben. Dies zeigt, dass sich mit signifikant geringerem Rechenaufwand ein Synopsenlayout maximaler Stichprobenraten gut annähern lässt.

### 6.3 Abschätzung der Synopsisgröße

Der zweite Teil der Experimente untersucht den Aufwand und die Genauigkeit für die Abschätzung der Synopsisgröße für gegebene Stichprobenraten. Da die Genauigkeit von der

<sup>5</sup> Es ist zu beachten, dass gilt: W-SR = K-SG, wenn  $\forall i, j : w_i = w_j$ ; im gleichen Sinne gilt auch: W-SG = K-SR, wenn  $\forall i, j : w_i = w_j$ .

Größe des Mehraufwandes abhängt, wurde K-SR als Allokationsschema mit mittlerem Mehraufwand gewählt. Hierfür wurden zunächst für verschiedene Datenformationen die entsprechenden Stichproben berechnet und davon ausgehend die Synopsisgrößen mit den verschiedenen Verfahren geschätzt. Die Ergebnisse werden in Abb. 14 präsentiert.

**Fremdschlüsselattributverteilung** Im ersten Experiment wurden nur die Tabellen Orders ( $O$ ) und Customer ( $C$ ) des modifizierten TPC-H-Schemas betrachtet. Die Schiefe der Häufigkeitsverteilung der Fremdschlüsselattributwerte variierte zwischen  $z = 0$  und  $z = 1$ . Dabei wird jedes Tupel aus  $C$  mindestens einmal referenziert. Da dies für  $z > 1$  nicht mehr gewährleistet ist, wurde  $z \leq 1$  gewählt, um ein Vermischen der Auswirkungen verschiedener Faktoren zu vermeiden. Abbildung 14a zeigt die Genauigkeit der Schätzung der Synopsisgröße. Da  $g^+$  für einfache Eltern-Kind-Beziehungen exakt ist, liegen entsprechend die Kurven von  $g$  und  $g^+$  übereinander. Mit  $g^-$  ist die geschätzte Synopsisgröße entsprechend der Größe des Mehraufwandes zu klein (siehe auch Abb. 11b), jedoch stets mindestens 90% der exakten Synopsisgröße. Weiterhin ist zu beobachten, dass die Schätzungen für  $g^-$  und  $g^=$  gleich sind,  $g^=$  also ebenfalls keinen Mehraufwand schätzt. Dies liegt am Verhältnis der Stichprobenraten und der Gleichverteilungsannahme; Details dazu sind in Abschn. 4.2 in [14] zu finden.

**Anteil unreferenzierter Tupel** Im nächsten Experiment wurde ähnlich vorgegangen, dieses Mal jedoch der An-

teil von nicht referenzierten Tupeln verändert. Dabei bedeutet ein Anteil von 25%, dass ein Viertel aller Kunden keine Bestellung getätigt hat. Die Schiefe der Fremdschlüsselattributverteilung wurde auf  $z = 0,5$  fixiert. Wie im letzten Experiment liefert  $g^+$  durch die einfache Eltern-Kind-Beziehung der Tabellen die exakte Synopsengröße, siehe Abb. 14b. Die Ungenauigkeit der Schätzung von  $g^-$  spiegelt die Höhe des Mehraufwandes für die verschiedenen Referenzierungsverhalten wider; die Schätzung schwankt zwischen 75% und 90% der exakten Größe. Da  $g^-$  den Anteil nicht referenzierter Tupel berücksichtigt, liefert diese Schätzung hier gute Werte (90–103% der exakten Größe).

*Anzahl direkter Vorgänger* In einem weiteren Experiment wurde der Einfluss der Anzahl der Elterntabellen untersucht. Hierfür wurde eine Kindtabelle mit 150 000 Tupeln und fünf Elterntabellen mit jeweils 1 500 000 Tupeln generiert (Customer-Orders-Verhältnis). Die Schiefe der Fremdschlüsselattributverteilung wurde jeweils auf  $z = 0,5$  gesetzt. In diesem Szenario wurde jeweils für eine steigende Anzahl an Elterntabellen die Synopsengrößen einer 1% Synopse geschätzt. Die Ergebnisse sind in Abb. 14c dargestellt. Erneut liefert  $g^+$  das exakte Ergebnis; die Schätzung von  $g^-$  überschätzt den Mehraufwand bis zu 25%, da die Kindtupel für jede Elterntabelle wieder als Kandidat für die Referenztabelle betrachtet werden können und somit mehrfach gezählt werden. Die Vernachlässigung des Mehraufwandes durch  $g^-$  hat für Kindtabellen mit vielen Elterntabellen deutliche Unterschätzungen (90–65% der exakten Synopsengröße) zur Folge, da die Referenzierungswahrscheinlichkeiten und damit die Wahrscheinlichkeiten für die Aufnahme in die Referenztabelle mit der steigenden Anzahl von Referenzen steigen.

*Anzahl Tabellen* Die letzten betrachteten Datenformationen variieren die Länge des Pfades im Schemagraphen. Dabei wurden die Tabellen Lineitem, Orders, Customer, Partsupp, Part und Supplier betrachtet und schrittweise in dieser Reihenfolge in das Schema aufgenommen. In jedem Schritt wurden die Stichprobenraten für eine 1% Synopse berechnet und für die Schätzung der Synopsengröße verwendet. Abbildung 14d zeigt die Ergebnisse für eine Schiefe der Fremdschlüsselattributverteilung von jeweils  $z = 0,5$ . Während der Mehraufwand für  $g^+$  für direkte Nachfolger noch korrekt berechnet wird, kommt es durch die Verwendung von  $q_i$  anstelle von  $pSel_{R_i}(t)$  (siehe Abschn. 4.3) für weitere Nachfolger entlang des Schemagraphen zu Unterschätzungen der Synopsengröße. Die geschätzten Werte liegen dabei stets bei mindestens 85% der exakten Werte. Bei der Verwendung von  $g^-$  und  $g^=$  fallen die Schätzungen auf bis zu 80% der exakten Werte.

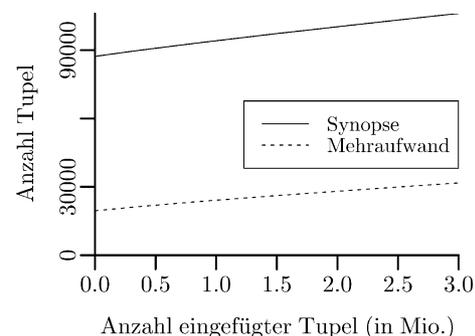
*Aufwand* Der Zeit für die Berechnung bzw. Abschätzung der Synopsengröße in Abhängigkeit der Anzahl der Tabellen im Schema ist in Abb. 14e dargestellt. Hierfür wurden

mehrere Messungen durchgeführt und die Laufzeiten gemittelt. Die exakte Berechnung der Synopsengröße dauerte dabei in Abhängigkeit der Anzahl der Tabellen zwischen 100 und 500 Sekunden, wohingegen die Schätzungen maximal 1,3 Sekunden dauerten. Die Berechnung von  $g^-$  dauert jeweils unter einer Millisekunde.

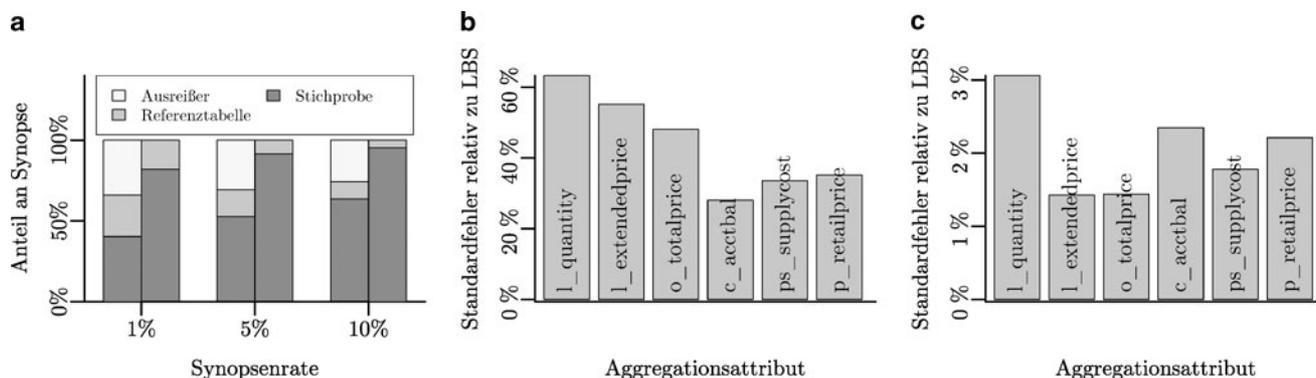
*Zusammenfassung* Aus diesen Experimenten ist zu erkennen, dass mit einer signifikanten Einsparung an Rechenzeit eine gute Abschätzung der resultierenden Synopsengröße für gegebene Stichprobenraten gegeben werden kann. Ist der Mehraufwand eines initialen Lesens und der Speicherung der Anzahl eingehender Referenzen pro Tupel akzeptabel, liefert  $g^+$  in vielen Fällen sehr genaue Schätzungen. Andernfalls kann auch mit wenig (Anzahl verschiedener Fremdschlüsselattributwerte) oder ohne Wissen eine zu meist gute Schätzung der Synopsengröße geliefert werden. Dies kann insbesondere für die Berechnung des V-OPT-Allokationsschemas zu einer erheblichen Reduzierung des Berechnungsaufwandes führen.

#### 6.4 Einfügen neuer Daten

Der nächste Teil der Experimente untersucht die Auswirkungen des Einfügens neuer Daten in die Faktentabelle auf die Synopsengröße und den Mehraufwand. Hierfür wurde zunächst eine Synopse der Größe 1% der Tabellen Lineitem, Orders, Customer, Partsupp, Part und Supplier mit dem Allokationsschema V-OPT angelegt. Anschließend wurden 3 Mio. neue Lineitem-Tupel generiert, deren Fremdschlüsselverteilung – entsprechend den vorhandenen (6 Mio.) Tupeln – einer Zipf-Verteilung mit  $z = 1$  folgen. Diese Tupel wurden schrittweise in die Lineitem-Tabelle eingefügt und die Synopse aktualisiert. Die Ergebnisse sind in Abb. 15 zu sehen: Die Größe der Synopse wächst entsprechend der Anzahl eingefügter Tupel mit; der Mehraufwand steigt dabei durch das ausschließliche Einfügen neu referenzierter Kindtupel in die Referenztabelle von 22,3% der Synopsengröße auf 29,9%.



**Abb. 15** Auswirkung des Anhängens neuer Daten an die Faktentabelle auf die Größe der Synopse bzw. des Mehraufwandes



**Abb. 16** Auswirkung der zusätzlichen Optimierung für Aggregationsattribute auf Synopsenlayout und Schätzfehler, **a** Synopsenlayout für  $z = 1$ , **b** Reduzierung des Schätzfehlers für  $z = 0,5$ , **c** Reduzierung des Schätzfehlers für  $z = 1$

Mit dieser einfachen Form der Wartung ist es möglich, eine einmal erstellte schemaweite Synopse inkrementell beim Einfügen neuer Tupel in die Faktentabelle zu aktualisieren. Die initial festgelegten Stichprobenraten bleiben erhalten, die einzelnen Stichproben statistisch korrekt.

### 6.5 Zusätzliche Optimierung für Aggregationsattribute

Im letzten Teil der Experimente wurden die Auswirkungen der zusätzlichen Optimierung für Aggregationsanfragen auf das Synopsenlayout und die Schätzgenauigkeit untersucht. Hierfür wurden Synopsen für folgende Tabellen und Aggregationsattribute berechnet.<sup>6</sup>

Tabelle	Aggregationsattribute
Lineitem	l_quantity, l_extendedprice
Orders	o_totalprice
Customer	c_acctbal
Partsupp	ps_supplycost
Part	p_retailprice
Supplier	s_acctbal

Die Schiefe der Häufigkeitsverteilung der Fremdschlüsselattributwerte sowie die der Werteverteilung der Aggregationsattribute wurde zunächst auf  $z = 0,5$  und in einem weiteren Versuchsaufbau auf  $z = 1$  fixiert. Anschließend wurden Synopsen der Größen 1% bis 10% berechnet. Das Synopsenlayout für eine Schiefe von  $z = 1$  ist in Abb. 16a dargestellt. Wie zu sehen ist, machen die Ausreißertabellen zwischen 25% und 35% der Synopse aus. Weiterhin ist zu erkennen, dass durch die höheren Referenzierungswahrscheinlichkeiten der Mehraufwand etwas ansteigt.

Die Abb. 16b und c zeigen, wie stark der Schätzfehler durch die zusätzliche Ausreißerbehandlung reduziert werden kann. Hierfür wurden die Standardfehler des geschätz-

ten Mittelwertes<sup>7</sup> für LBS und LBS mit Ausreißerbehandlung (jeweils der Größe 5%) berechnet und ins Verhältnis gesetzt. Für Daten mit wenigen Ausreißern ( $z = 0,5$ ) sinkt der Standardfehler der Schätzers durch die Ausreißerbehandlung auf 63,3% bis 28,0%. Bei stärker verzerrten Daten ( $z = 1$ ) kann der Standardfehler auf 3,1% bis 1,4% reduziert werden.

Diese Experimente verdeutlichen klar den Vorteil der Erweiterung von Linked-Bernoulli-Synopsen um die Ausreißerbehandlung aus [20]: Der Schätzfehler für Aggregationsanfragen mit Fremdschlüsselverbundoperationen kann signifikant gesenkt werden.

## 7 Verwandte Arbeiten

Im Bereich der näherungsweise Anfrageverarbeitung gibt es eine Vielzahl von Synopsenformen. Im relationalen Kontext sind dabei Histogramme, Wavelets und Stichproben weit verbreitet. Histogramme haben ihren Ursprung in der Anfrageoptimierung und stellen eine Näherung der Datenverteilung einzelner oder mehrerer Attribute dar; sie sind weit erforscht und Bestandteil nahezu aller kommerzieller Datenbanksysteme. Der Einsatz von Histogrammen für die näherungsweise Anfrageverarbeitung wird in [16, 19] diskutiert. Allerdings lassen sich für Histogramme keine Fehlergrenzen, wie beispielsweise Konfidenzintervalle, berechnen. Die näherungsweise Ergebnisse sind damit kaum interpretierbar.

Wavelets nähern den Verlauf einer Datenverteilung durch die Mittelung wertemäßig ähnlicher Bereiche an. Deterministischen Wavelets, wie [5, 13, 24] fehlen ebenso wie Histogrammen Fehlergrenzen und damit Aussagen zur Qualität des geschätzten Ergebnisses. Mit probabilistischen Wavelets [10, 12] sind Fehlergrenzen möglich, die

<sup>6</sup> Es wurde das  $\mathcal{M}_{AVG}$ -Fehlermaß verwendet, siehe [20].

<sup>7</sup> entspricht der Wurzel aus der Varianz des geschätzten Mittelwertes, vgl. Gl. 3

teure Rücktransformationen der Daten bzw. die aufwändige Anpassung der relationalen Operatoren an die Wavelet-Koeffizienten schränken die Einsetzbarkeit von Wavelets in Datenbanksystemen dennoch stark ein.

Stichproben stellen eine sehr einfache und weit verbreitete Synopsenform dar. Sie sind allgegenwärtig in Szenarien der Massendatenauswertung; ihr Anwendungsbereich umfasst die näherungsweise Anfrageverarbeitung [2, 3, 6], die Anfrageoptimierung [4, 7], Datenanalysen [18, 22] sowie die Stromverarbeitung [17]. Ein großer Vorteil von Stichproben gegenüber anderen Synopsenformen ist ihre Anwendbarkeit auf einem breiten Anfragespektrum sowie die Möglichkeit der Berechnung probabilistischer Fehlergrenzen.

Die meisten Arbeiten im Bereich der stichprobenbasierten, näherungsweise Anfrageverarbeitung befassen sich mit Stichprobenverfahren auf Tabellenebene. Eine Stichprobe auf Tabellenebene repräsentiert eine einzelne Tabelle einer Datenbank. Hier liegt der Fokus der Forschung auf Stichprobenverfahren, welche für verschiedene spezielle Anfragetypen wie Aggregationsanfragen [6, 20] oder Gruppierungsanfragen [2, 3] optimiert sind. Wie in Abschn. 5.2 gezeigt, können diese Verfahren mit den in diesem Artikel betrachteten Linked-Bernoulli-Synopsen kombiniert werden.

Andere Forschungsarbeiten betrachten Stichprobenverfahren auf Schemaebene. Stichproben dieser Ebene repräsentieren eine Menge von Tabellen und den Beziehungen zwischen ihnen. Hierzu zählen die von Acharya et al. vorgeschlagenen *Join-Synopsen* [1] sowie die in diesem Artikel näher betrachteten Linked-Bernoulli-Synopsen [14]. Linked-Bernoulli-Synopsen erweitern Join-Synopsen durch Korrelation der einzelnen Stichproben und der damit verbundenen Minimierung des Speicherbedarfes. Eine weitere Zusammenfassung auf Schemaebene bietet die *Tuple-Graph-Datenstruktur* von Spiegel und Polyzotis [21], welche komplexe Beziehungen zwischen den Tupeln widerspiegelt. Ebenso wie die *Probabilistic Relational Models* [15], welche statistische Abhängigkeiten zwischen Attributen verschiedener Tabellen modellieren, fokussieren sich diese Synopsen auf die Selektivitätsschätzung komplexer Anfragen und können nicht für die näherungsweise Anfrageverarbeitung verwendet werden.

## 8 Zusammenfassung

Linked-Bernoulli-Synopsen bieten eine hervorragende Möglichkeit, auch näherungsweise Anfragen mit Fremdschlüsselverbundoperationen speichereffizient zu unterstützen. Insbesondere in Schemata mit mehreren großen Tabellen, wie beispielsweise Galaxy-Schemata, kann mit diesem Verfahren die interaktive Analyse von Daten durch approxi-

mative Anfrageverarbeitung und damit verbunden kurzen Antwortzeiten erheblich profitieren.

In diesem Artikel wurden wesentliche Aspekte für den praktischen Einsatz von Linked-Bernoulli-Synopsen in Data-Warehouse-Umgebungen dokumentiert. Dies betrifft zum einen die Fragestellung der Speicherplatzaufteilung für die einzelnen Tabellensynopsen schemaweiter Synopsen; hierfür wurden verschiedene Allokationsschemata vorgestellt und deren Auswirkungen auf den Mehraufwand untersucht. Zum anderen wurden neben diesem Synopsenerstellungsaspekt auch Thematiken der Synopsenverwendung adressiert; für Änderungen der Datenbasis wurde eine Möglichkeit der inkrementellen Wartung präsentiert; durch eine Erweiterung von Linked-Bernoulli-Synopsen um die Ausreißerbehandlung aus [20] lässt sich der Schätzfehler von Aggregationsanfragen deutlich reduzieren. Mit einer Vielzahl von Experimenten wurde gezeigt, dass Linked-Bernoulli-Synopsen und die in diesem Artikel diskutierten Verfahren vielversprechend für den Einsatz in Data-Warehouse-Szenarien sind.

## Literatur

1. Acharya S, Gibbons PB, Poosala V, Ramaswamy S (1999) Join Synopses for Approximate Query Answering. In: SIGMOD, S 275–286
2. Acharya S, Gibbons PB, Poosala V (2000) Congressional Samples for Approximate Answering of Group-By Queries. In: SIGMOD, S 487–498
3. Babcock B, Chaudhuri S, Das G (2003) Dynamic Sample Selection for Approximate Query Processing. In: SIGMOD, S 539–550
4. Brown PG, Haas PJ (2003) BHUNT: Automatic Discovery of Fuzzy Algebraic Constraints in Relational Data. In: VLDB, S 668–679
5. Chakrabarti K, Garofalakis MN, Rastogi R, Shim K (2000) Approximate Query Processing Using Wavelets. In: VLDB, S 111–122
6. Chaudhuri S, Das G, Datar M, Motwani R, Narasayya VR (2001) Overcoming Limitations of Sampling for Aggregation Queries. In: ICDE, S 534–544
7. Chaudhuri S, Das G, Srivastava U (2004) Effective Use of Block-Level Sampling in Statistics Estimation. In: SIGMOD, S 287–298
8. Chaudhuri S, Dayal U (1997) An Overview of Data Warehousing and OLAP Technology. SIGMOD Rec 26(1):65–74
9. Chaudhuri S, Motwani R, Narasayya V (1999) On Random Sampling over Joins. In: SIGMOD, S 263–274
10. Deligiannakis A, Garofalakis MN, Roussopoulos N (2005) A Fast Approximation Scheme for Probabilistic Wavelet Synopses. In: SSDBM, S 243–252
11. Salley CT, Codd EF, Codd SB (1993) Providing OLAP to User-Analysts: An IT Mandate
12. Garofalakis M, Gibbons PB (2004) Probabilistic Wavelet Synopses. ACM Trans Database Syst 29(1):43–90
13. Garofalakis M, Kumar A (2005) Wavelet Synopses for General Error Metrics. ACM Trans Database Syst 30(4):888–928
14. Gemulla R, Rösch P, Lehner W (2008) Linked Bernoulli Synopses: Sampling Along Foreign-Keys. In: SSDBM, S 6–23
15. Getoor L, Taskar B, Koller D (2001) Selectivity Estimation using Probabilistic Models. In: SIGMOD, S 461–472

16. Ioannidis YE, Poosala V (1999) Histogram-Based Approximation of Set-Valued Query-Answers. In: VLDB, S 174–185
17. Johnson T, Muthukrishnan S, Rozenbaum I (2005) Sampling Algorithms in a Stream Operator. In: SIGMOD, S 1–12
18. Kollios G, Gunopulos D, Koudas N, Berchtold S (2001) An Efficient Approximation Scheme for Data Mining Tasks. In: ICDE, S 453–462
19. Poosala V, Ganti V, Ioannidis YE (1999) Approximate Query Answering Using Histograms. IEEE Data Eng Bull 22:5–14
20. Rösch P, Gemulla R, Lehner W (2008) Designing Random Sample Synopses with Outliers. In: ICDE, S 1400–1402
21. Spiegel J, Polyzotis N (2006) Graph-Based Synopses for Relational Selectivity Estimation. In: SIGMOD, S 205–216
22. Toivonen H (1996) Sampling Large Databases for Association Rules. In: VLDB, S 134–145
23. Tuy H (2000) Monotonic Optimization: Problems Solution Approaches. SIAM J Opt 11(2):464–494
24. Vitter JS, Wang M (1999) Approximate Computation of Multidimensional Aggregates of Sparse Data Using Wavelets. In: SIGMOD, S 193–204

**Philipp Rösch** studierte an der TU Ilmenau Informatik. Er ist seit 2005 wissenschaftlicher Mitarbeiter an der Technischen Universität Dresden am Lehrstuhl für Datenbanken unter der Leitung von Prof. Lehner. Sein Forschungsschwerpunkt ist die näherungsweise Anfrageverarbeitung von Datenbank-anfragen.

**Prof. Dr.-Ing. Wolfgang Lehner** (geb. 1969) studierte Informatik an der Universität Erlangen-Nürnberg, wo er 1998 promovierte und 2001 habilitierte. Seit Oktober 2002 ist er Inhaber des Lehrstuhls für Datenbanken an der Technischen Universität Dresden. Er war Gastwissenschaftler am IBM Almaden Research Center in San Jose, bei Microsoft Research in Redmond und bei SAP in Walldorf. Sein Forschungsschwerpunkt liegt in der Entwicklung von Architekturen von Datenbank- und Informationssystemen zur Unterstützung datenintensiver Anwendungen und Prozesse.