

Integration von Stichproben- verfahren in ein relationales DBMS

Bernhard Jäcksch

Studentenprogramm BTW 2007, Aachen

Lehrstuhl für Datenbanken | Institut für Systemarchitektur | TU Dresden

- **Motivation**
- **Grundlagen**
 - Stichproben
 - Derby
- **Zielstellung**
- **Vorstellung des entwickelten Frameworks**
 - Übersicht
 - Erläuterung an einem Beispiel
- **Zusammenfassung**

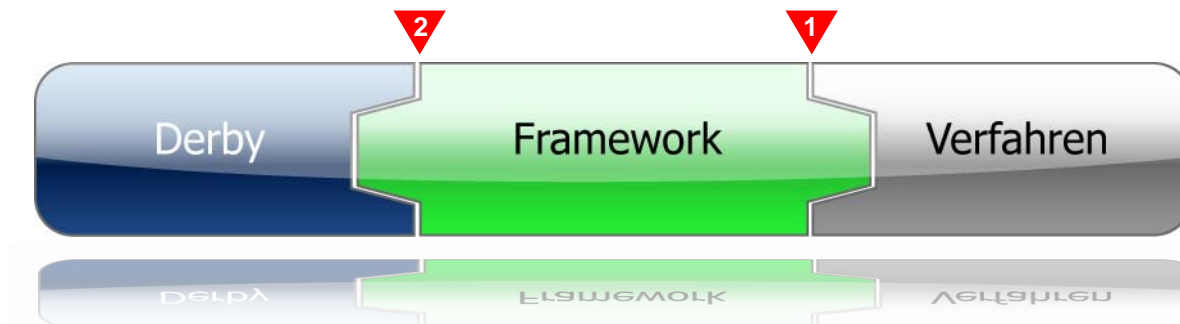
- **Daten werden überall erhoben → große Datenmengen (Data-Warehouse-Systeme)**
- **Daten liefern Informationen über bzw. Sichten auf beobachtete Prozesse → Decision-Support-Systeme**
- **Interaktive Benutzung der DSS wünschenswert ↗ Menge der Daten**
- **Näherungsweise Ergebnisse akzeptabel → Stichproben**

- **Repräsentative Teilmenge einer Originaldatenmenge**
- **Nur ein Bruchteil der Originalgröße**
- **Materialisierte Stichproben werden separat abgelegt**
- **Anfrageergebnisse lassen sich mit Hilfe von Stichproben approximieren**
- **Aussagen über die Qualität der Ergebnisse in Form von Konfidenzintervallen möglich**
- **Unterschiedliche Verfahren, um Stichproben zu erstellen**
- **Komplexere Verfahren für bestimmte Anfragearten optimiert (Gruppierung, Aggregation, Fremdschlüsselverbund)**

- **Open-Source Datenbank der Apache Software Foundation**
- **Komplett in Java**
- **Kleines, kompaktes Archiv (ca. 3 MB)**
- **Möglichst konform zum SQL-Standard**
- **Derby/S Prototyp als Grundlage der Arbeit**

- **Plugin-Schnittstelle zur einfachen Integration von Stichprobenverfahren**
- **Geringer Einarbeitungsaufwand in Derby-Interna**
- **Möglichkeit Verfahren miteinander zu kombinieren**
- **Berücksichtigung der von Derby/S vorgesehenen Wartungsmechanismen**
- **Derby/S soll als Plattform für Demonstration und Tests neuer Stichprobenverfahren dienen**

- **Plugin-Schnittstelle mit deren Hilfe möglichst viele Verfahren in Derby/S eingebunden werden können (1)**
- **Aufgaben des Frameworks**
 - Erzeugung
 - Wartung
 - Verwendung von Stichproben
 - Schnittstelle zu Derby **(2)**



- Tabelle *sales*

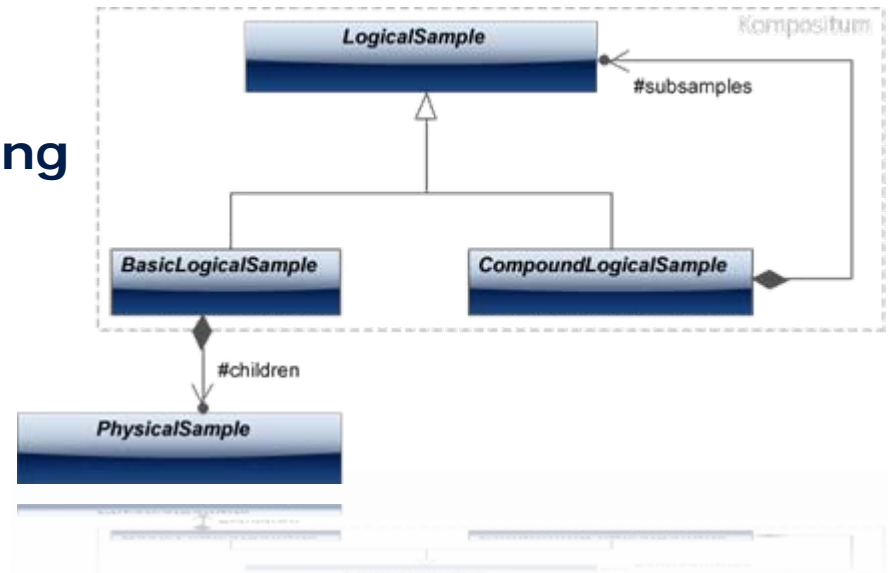
- $|sales| = 38$

S_ID	S_NAME	S_PRICE	S_YEAR	S_COUNTRY
1	Order 0001	5678.00	1998	Spain
2	Order 0002	11599.05	1998	France
3	Order 0003	12356.10	1998	Germany
4	Order 0004	45236.15	1998	Austria
5	Order 0005	8649.20	2006	France
...

- **Neuer SQL-Befehl um Stichproben zu erstellen**
- **Alle Parameter werden in Derby Systemkatalogen gespeichert**
- **Auswahlstrategie für Stichprobenverfahren austauschbar**

```
CREATE SAMPLE s1 AS  
(SELECT * FROM sales)  
  
OF SIZE 8 TUPLES  
  
FOR GROUP BY s_year, s_country  
  
MANAGED BY SYSTEM  
  
REFRESH IMMEDIATE
```

- **Viele komplexere Verfahren bestehen aus mehreren Teilstichproben**
- **Abbildung auf Entität in der Datenbank → Tabelle**
- **Physische Stichproben repräsentieren eine Tabelle**
- **Logische Stichproben stellen die nach außen hin sichtbare Gesamtstichprobe dar**
- **Baumartige Hierarchiebeziehung**



- Stichprobe *S1*

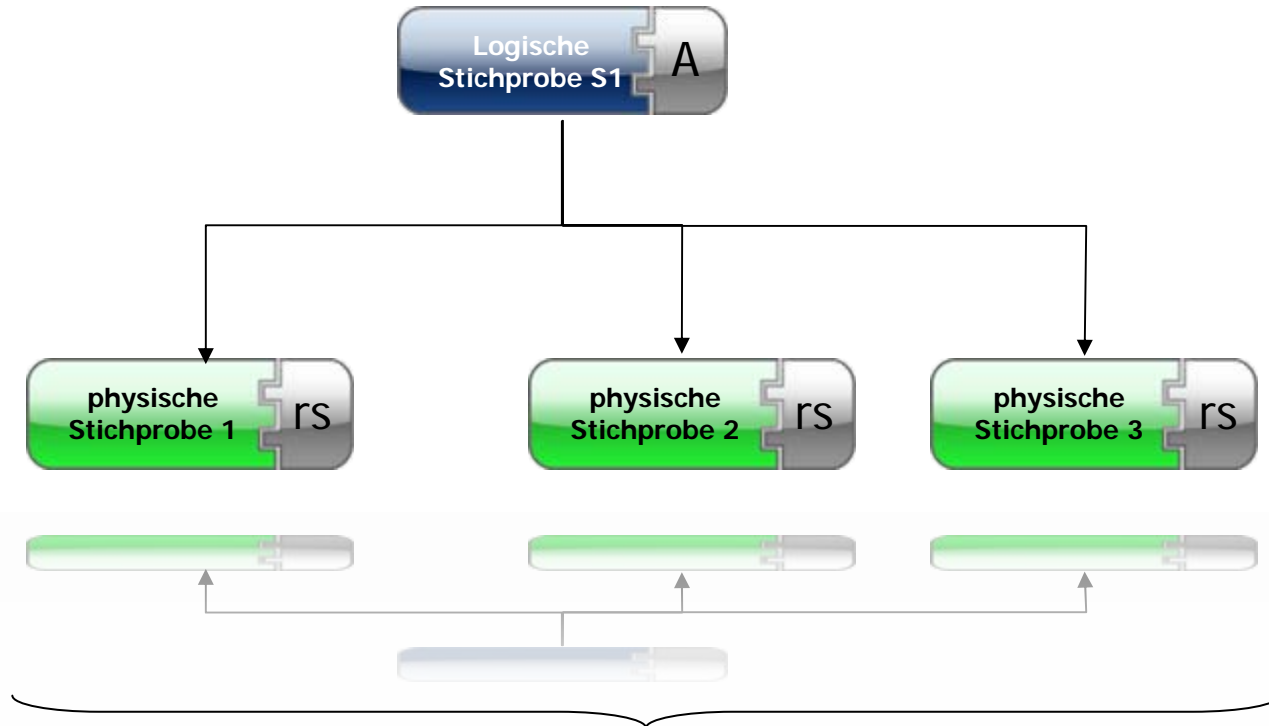


Abbildung auf Tabellen

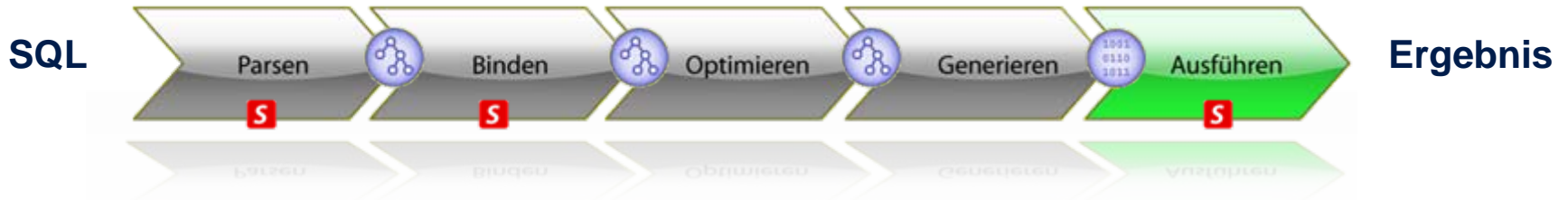
- **Approximative Anfrage wird auf Original-Relationen formuliert**
- **Framework schreibt Anfrage selbstständig um**

```
SELECT s_year, SUM(s_price), AVG(s_price)
FROM sales
GROUP BY s_year
```

} **exakt**











```
SELECT SOME s_year, ~SUM(s_price), ~AVG(s_price)
FROM sales
GROUP BY s_year
```

} **approximativ**



- SQL Anfrage wird vom Parser geparst → Baum
- Transformieren des Baumes (approx. Anfrage)
- Binden → sind alle Spalten/Relationen korrekt
- Anfrageplan wird optimiert
- Generieren einer Java-Bytecode Klasse → enthält Plan zur Erzeugung der Ergebnismenge
- Bytecode wird ausgeführt und liefert das Ergebnis

Derby/S Demo - SIGMOD'06

Console | Compile plan | Execution plan | Compare results | Refresh sample | System catalog | Sample catalog | Diagnosis | History | Configuration | About

```

SELECT s_year, SUM(s_price), AVG(s_price) FROM sales GROUP BY s_year
  
```

S_YEAR	2	3
1998	74869.3	18717.325
2006	295081.85	8678.8779

2 row(s) selected.
Total time: 0.020s

```

SELECT SOME s_year, ~SUM(s_price), ~AVG(s_price) FROM sales GROUP BY s_year
  
```

S_YEAR	2	3
1998	74869.3	18717.325
2006	238003.5475	8351.0016

2 row(s) selected.
Total time: 0.040s

Used sample 'DERBY.S1' (algorithm SmallGroupSample) to answer approximate query.

```

SELECT SOME s_year, ~SUM(s_price), ~AVG(s_price) FROM sales GROUP BY s_year
  
```

S_YEAR	2	3
2006	196591.8125	5173.4687

1 row(s) selected.
Total time: 0.030s

Used sample 'DERBY.RS1' (algorithm ReservoirSample) to answer approximate query.

Connection to 'belegDB' established.

- **Einbindung stratifizierender Verfahren**
- **Einbindung von Online-Sampling**
- **Framework für Schätzer (Estimator)**
- **Schnittstelle zu Workload-Informationen**

- **Integration neuer Verfahren durch Modularisierung wesentlich vereinfacht → Grundlage für einheitliche Stichprobenintegration**
- **Voraussetzung für Integration komplexer Verfahren**
- **Quellcode-Komplexität des bisherigen Prototyps verringert**
- **Kontinuierliche Weiterentwicklung**
 - Integration neuer Verfahren
 - Verwendung des Framework
- **→ Identifizierung und Verbesserung von Schwachstellen**



- **Fragen, Anregungen bzw. Kritik ?**

- **Vielen Dank!**