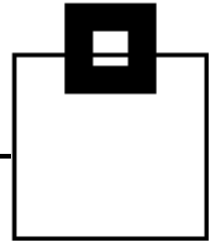


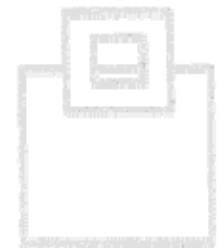
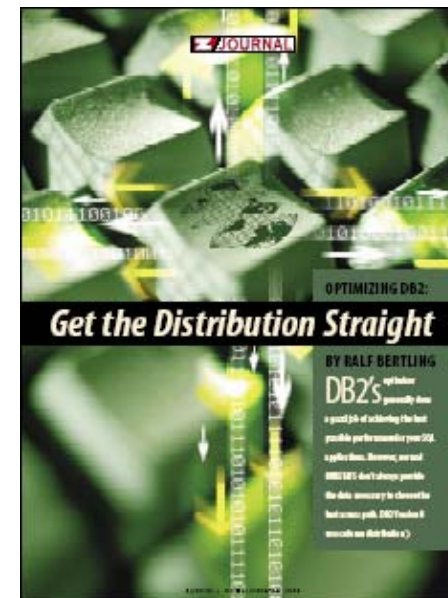
# Weniger, bessere RUNSTATS

---



## Achieving better Statistics with fewer, but better RUNSTATS

Ralf Bertling – Forschung und Entwicklung  
SOFTWARE ENGINEERING GMBH

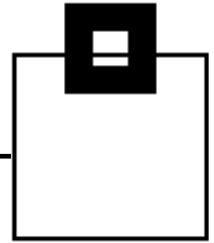


Frühjahrestagung DB2 Systemverwaltung Nord/Süd  
Düsseldorf – 17.- 18. Mai 2006

SOFTWARE ENGINEERING

# SOFTWARE ENGINEERING – Stand 2006:

---



## Jubiläum:



“We are in the Tools Business” – seit 1986

## Current seminars & Workshops:

- DB2 Migration to V8 – Be Prepared



## Current development highlights:

- DB2 V8  **EarlyPrecheck** ✓ (GA seit Januar 2006)
- DB2 V9 Tests & Entwicklung (Forschung seit Feb. 2005)
- Nahrung für den Optimizer – "Best in, Best Out"
-  **Recovery HealthCheck**



SOFTWARE ENGINEERING



# Agenda

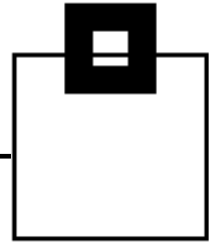
---

- Column distribution stats: Nahrung für den Optimizer
- Kosten-Nutzen Abwägung für RUNSTATS
- Vermeidung überflüssiger (Verteilungs-)Statistiken
- Ein praktikabler Ansatz auf Basis von COLGROUP bzw. FREQVAL
- Den Einfluss der Statistiken auf JOIN-Zugriffspfade prüfen
- Zusammenfassung & Ausblick
- Fragen & Antworten

SOFTWARE ENGINEERING

# Grundlagen: RUNSTATS, RTS, Optimizer

---

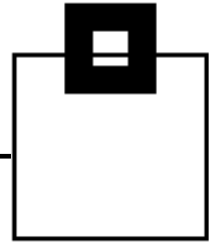


- Wahl des Zugriffspfads beruht auf Statistiken
  - Abhängigkeit (optimaler) Zugriffspfade von Daten(mengen)
  - Rechengrößen sind i.d.R. Filterfaktoren und nötige I/Os
  - Einfluss auf JOIN-Reihenfolge, -methode, Index-Benutzung...
- Aufstieg und Fall von RUNSTATS für die Datenbank-Wartung
  - DB2 V1 – V6: RUNSTATS als Auslöser für REORG & COPY
  - V7+: RTS für Wartungsjobs, RUNSTATS für den Optimizer
  - V7, V8 & V9: Neue RUNSTATS-Optionen: Was ist wo sinnvoll?

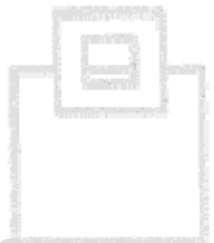
SOFTWARE ENGINEERING

# Agenda

---



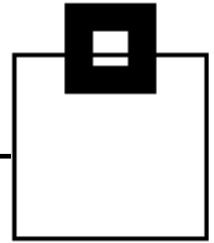
- Column distribution stats: Nahrung für den Optimizer
- **Kosten-Nutzen Abwägung für RUNSTATS**
- Vermeidung überflüssiger (Verteilungs-)Statistiken
- Ein praktikabler Ansatz auf Basis von COLGROUP bzw. FREQVAL
- Den Einfluss der Statistiken auf JOIN-Zugriffspfade prüfen
- Zusammenfassung & Ausblick
- Fragen & Antworten



SOFTWARE ENGINEERING

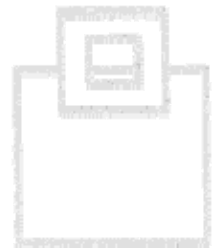
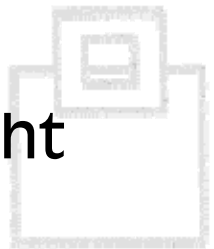
# Auswirkungen veralteter Statistiken

---



## Prioritäten bei der Aktualisierung

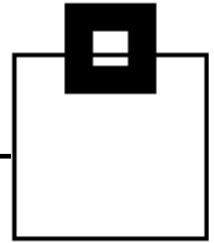
- Tabelle nach RUNSTATS drastisch verkleinert: Harmlos
- Tabelle nach RUNSTATS drastisch gewachsen: Sehr schlecht
- Veränderung der CLUSTERRATIO: Schlecht
- Beispiel: "Volatile" Tabellen (Größe schwankt stark)



SOFTWARE ENGINEERING

# Rat: RUNSTATS & REBIND durchführen?

---



- Ratschlag (für DB2 V7, V8 und wahrscheinlich V9)
  - RTS nutzen, um Nutzen & Risiken eines Utilities abzuwägen
  - Vergleich der wahrscheinlichen Änderungen und Kosten
  - Entsprechende Generierung von Utilities
  - Vorkehrungen für den Fall "gefährlicher" Statistiken
- Aber: Schrumpfende Tabellen nicht vergessen



SOFTWARE ENGINEERING

# Einschub: Korrelation



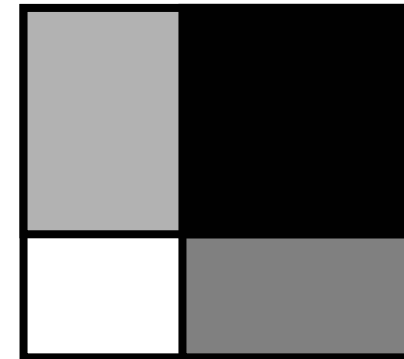
Daten (weiß)



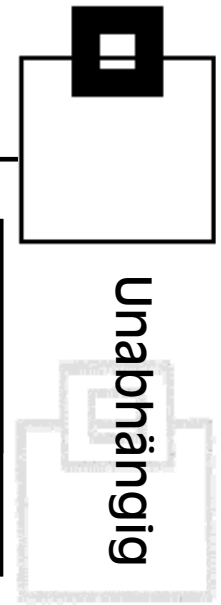
Bedingung X



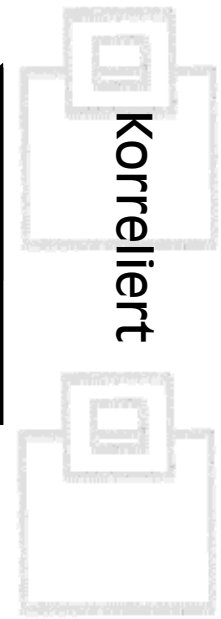
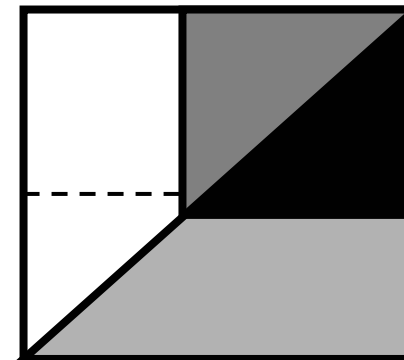
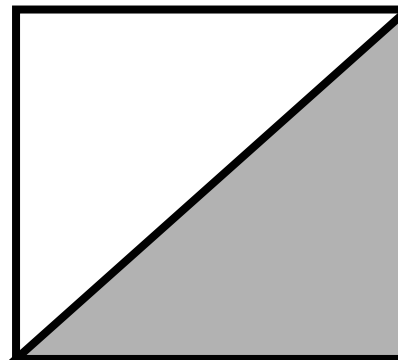
Bedingung Y



Ergebnis (AND)



unabhängig

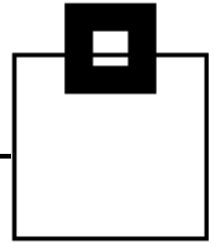


korreliert

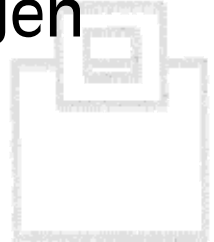
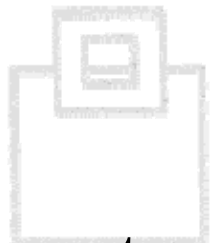
SOFTWARE ENGINEERING

# Verteilungsstat.: korrelierende Daten

---



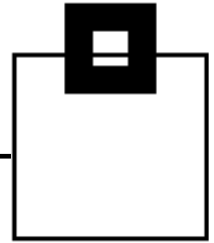
- korrelierende Daten: Ein Grund für Kopfschmerzen
  - "Warum läuft der Cursor so lange?"
  - Schlechter Zugriffspfad aufgrund falscher Annahmen
- Verteilungsstatistiken: Neu in V8 und im Optimizer verbessert
- Zeigt ungleichmäßige/korrelierte Daten (z.B. PLZ/Stadt)
  - Ändert falsche Standardannahmen im Optimizer
  - Nutzen nur bei erheblichen Abweichungen (vom Default)
  - Meistens führt Korrelation zu größeren Ergebnismengen



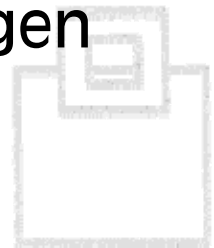
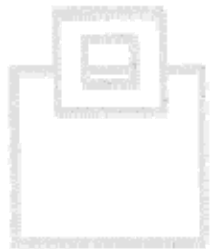
SOFTWARE ENGINEERING

# Verteilungsstat.: Futter für Optimizer

---



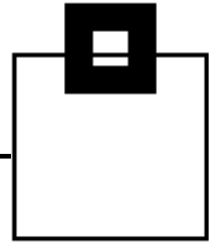
- Wie helfen Verteilungsstatistiken?
  - korrelierende Daten
  - ungleichmäßig verteilte Daten
- Betrachtung von JOINS am wichtigsten
  - Bei SQL mit nur einer Tabelle keine großen Auswirkungen
  - Bei JOINS Einfluss auf Reihenfolge und Methode
  - Kostenschätzungen scheinen schlechter, sind aber genauer



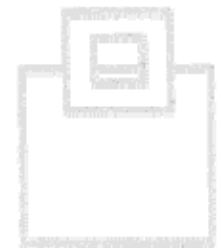
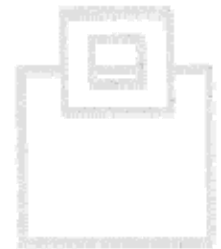
SOFTWARE ENGINEERING

# Kosten/Nutzen-Abwägung f. RUNSTATS

---

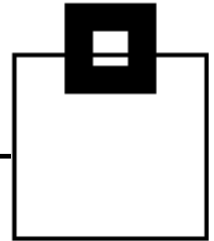


- Offensichtliche Kosten: Ausführung des RUNSTATS
  - Statistiken auf Basis von RTS erzeugen
  - Option "SAMPLE" geeignet verwenden
  - Option "FREQVAL" bei führenden Index-Spalten
  - Option "COLGROUP" nach Bedarf in DB2 V8
- weniger offensichtliche Kosten
  - Der dynamic statement cache geht verloren
  - Katalogaktivität am Ende der Ausführung
  - Zugriff kollidiert mit "Neubau" des Dyn. Stmt. Cache



# Abwägung: Versteckte RUNSTATS-Kosten

---



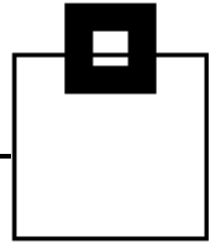
- Versteckte Kosten:
  - Von Hand zu reparierende RTS nach mass deletes
  - Verarbeitung der SYSSTATS Tabellen bei BIND & PREPARE
  - Schlechte Zugriffspfade durch
    - ungenaue Statistik-Daten
    - unvollständige Statistik-Daten
    - alte Statistik-Daten



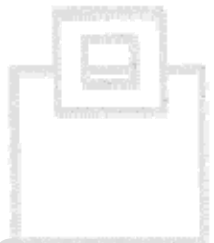
SOFTWARE ENGINEERING

# Agenda

---



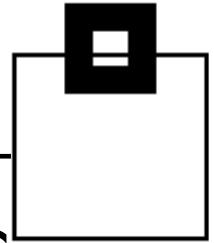
- Column distribution stats: Nahrung für den Optimizer
- Kosten-Nutzen Abwägung für RUNSTATS
- **Vermeidung überflüssiger (Verteilungs-)Statistiken**
- Ein praktikabler Ansatz auf Basis von COLGROUP bzw. FREQVAL
- Den Einfluss der Statistiken auf JOIN-Zugriffspfade prüfen
- Zusammenfassung & Ausblick
- Fragen & Antworten



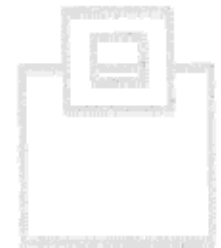
SOFTWARE ENGINEERING

# Vermeidung überflüssiger RUNSTATS-Kosten

---



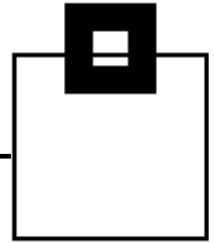
- Sparpotential durch Vermeidung überflüssiger RUNSTATS
  - RUNSTATS Ausführung
  - Locks auf dem DB2-Katalog
  - Mini-BINDs bei dynamischem SQL



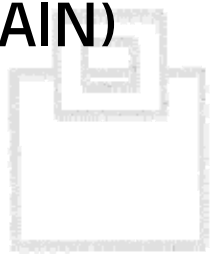
SOFTWARE ENGINEERING

# Vermeidung überflüssiger RUNSTATS

---



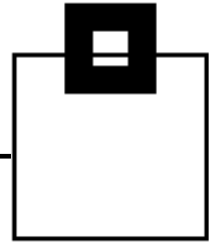
- RUNSTATS nur nach Veränderungen der RTS-Daten
- Statistiken, die nichts bringen löschen (Prüfung mit EXPLAIN)
- Option "COLGROUP" nur für Tabellen aus JOINS
- RUNSTATS mit "COLGROUP" selten wiederholen
- RUNSTATS mit "COLGROUP" wiederholen
- Keinen index nur für die Option "FREQVAL" anlegen
- Option "FREQVAL" wenn möglich nutzen



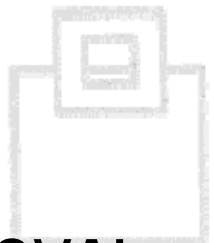
SOFTWARE ENGINEERING

# Agenda

---



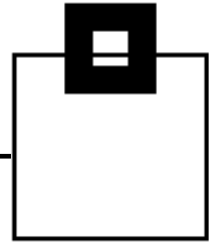
- Column distribution stats: Nahrung für den Optimizer
- Kosten-Nutzen Abwägung für RUNSTATS
- Vermeidung überflüssiger (Verteilungs-)Statistiken
- **Ein praktikabler Ansatz auf Basis von COLGROUP bzw. FREQVAL**
- Den Einfluss der Statistiken auf JOIN-Zugriffspfade prüfen
- Zusammenfassung & Ausblick
- Fragen & Antworten



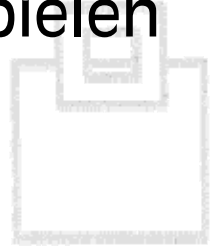
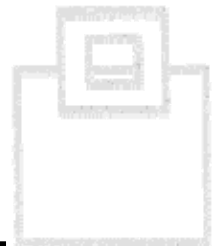
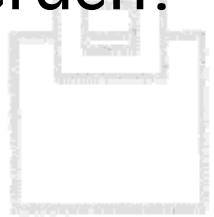
SOFTWARE ENGINEERING

# Ein praktikabler Ansatz...

---



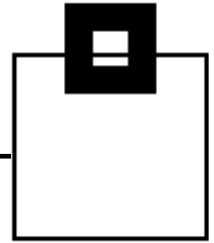
- Kandidaten für erweiterte RUNSTATS finden
- Kann FREQVAL vor/anstelle von COLGROUP verwendet werden?
- Alte Statistikdaten und alten Zugriffspfad sichern
- RUNSTATS ausführen
- Zugriffspfade mittels dynamischem EXPLAIN vergleichen
  - Vorsicht bei volatilen Tabellen
  - Scheinbar verschlechterte Zugriffspfade evtl. verwenden
  - Alte Statistiken bei gleichem Zugriffspfad wieder einspielen



SOFTWARE ENGINEERING

# Ein praktikabler Ansatz... (Teil 2)

---

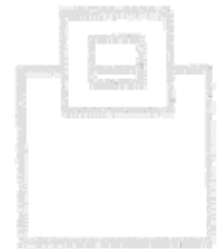
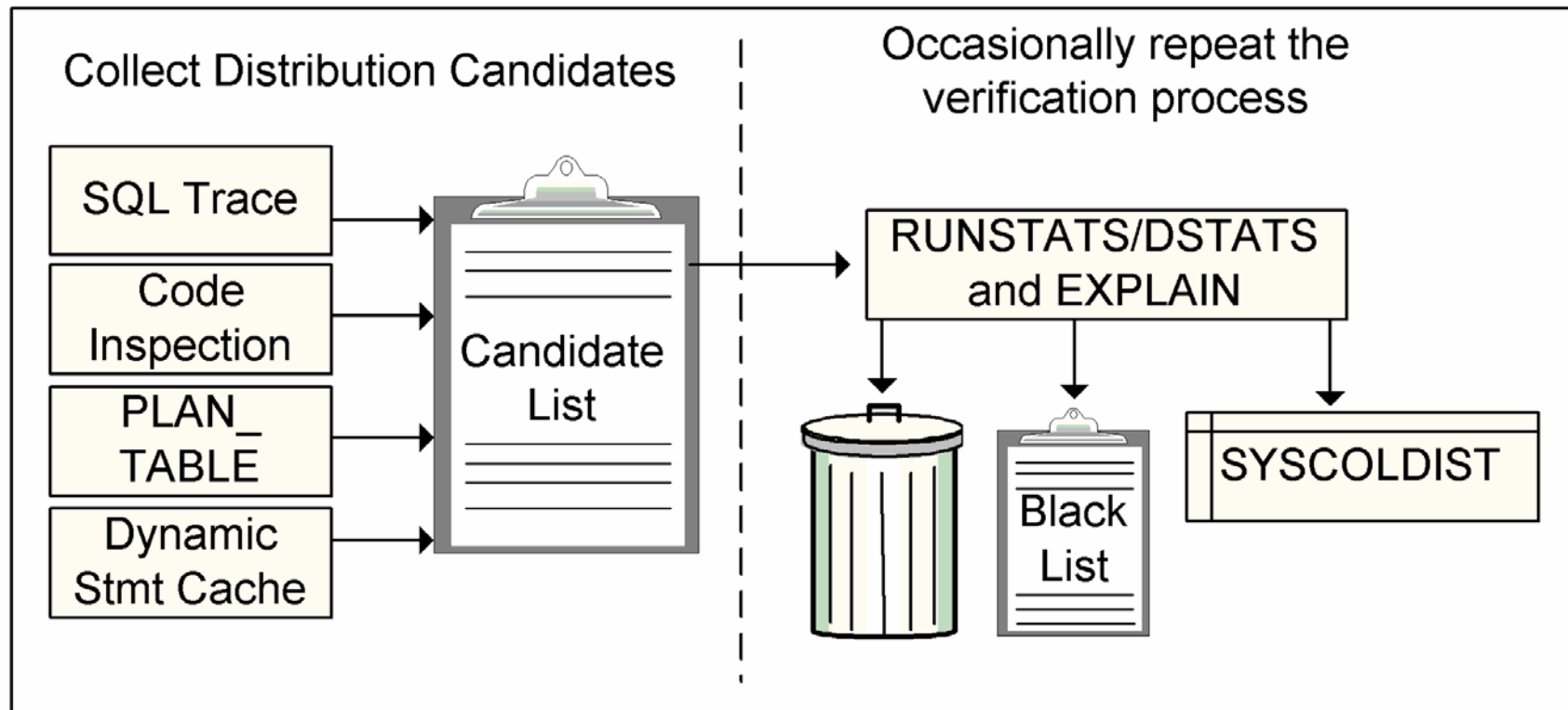
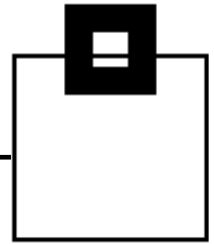


- Etwas Buchführung unumgänglich
  - Schwarze Liste gegen zahlreiche Versuche pro Tabelle
  - Objekteigenschaften speichern (Änderungen erkennen)
    - Verteilungssstatistiken gelegentlich aktualisieren
    - RUNSTATS bei gravierenden Veränderungen wiederholen
    - Schwarze Liste bei gravierenden Änderungen aufräumen
  - Kosten und Nutzen des Verfahrens bewerten
- Statistiken regelmäßig auf Konsistenz prüfen



SOFTWARE ENGINEERING

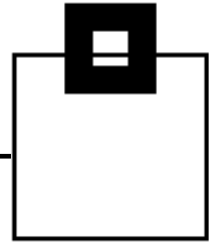
# Ein praktikabler Ansatz – Überblick



SOFTWARE ENGINEERING

# Kandidatenauswahl oder "it depends"

---

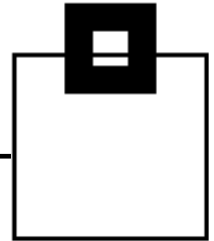


- Im Idealfall ... SQL-Code und alle Tabellen bekannt
  - Erstellung der Kandidatenliste von Hand
  - Statistiken mittels SPUIFI einfügen
- Zugriff auf Code & Laufzeitdaten (Profiling) möglich
  - Suche SQLs mit JOIN-Bedingungen über mehrere Spalten
  - Kosten laut EXPLAIN mit tatsächlicher Laufzeit vergleichen
  - Suche entsprechende JOIN-Bedingungen im dyn. stmt cache
- Trace durchführen, um eine gewichtete Liste von Spaltengruppen zu erhalten (z.B. bei SAP-Systemen)
- Berücksichtigung der schwarzen Liste



# FREQVAL, COLGROUP oder SQL?

---

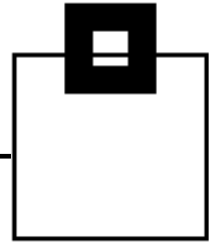


- Kandidaten sind interessant, nicht immer korrelierend
- Option FREQVAL ist viel schneller als COLGROUP
  - Bei führenden Indexspalten nutzen, keinen Index erstellen
  - Nutzung von FREQVAL möglich, um Auswahl einzugrenzen
- Spezielle SQL (SELECT COUNT(\*), ... GROUP BY ... WITH UR)
  - Keine Gefahr durch volatile Tabellen
  - Evtl. bessere Indexnutzung, wo FREQVAL ausscheidet
  - Direkte Überprüfung von Filterfaktoren aus EXPLAIN
  - Bei teuren JOINS sind einige Spalten sowieso im Index

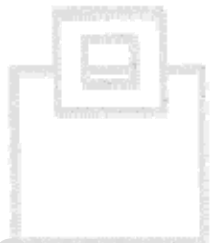


# Agenda

---



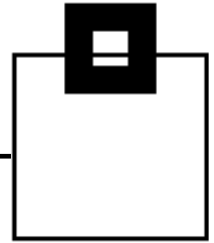
- Column distribution stats: Nahrung für den Optimizer
- Kosten-Nutzen Abwägung für RUNSTATS
- Vermeidung überflüssiger (Verteilungs-)Statistiken
- Ein praktikabler Ansatz auf Basis von COLGROUP bzw. FREQVAL
- **Den Einfluss der Statistiken auf JOIN-Zugriffspfade prüfen**
- Zusammenfassung & Ausblick
- Fragen & Antworten



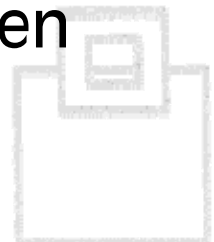
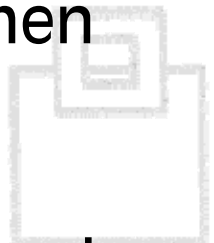
SOFTWARE ENGINEERING

# Einfluss ... auf Zugriffspfade prüfen

---



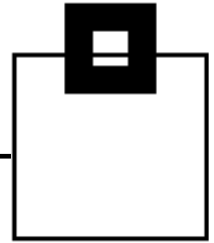
- EXPLAIN der SQL vor & nach dem erweiterten RUNSTATS
  - Vorher verwendet der Optimizer ggfs. falsche Annahmen
  - Zugriffspfad ermitteln ohne zu binden?
    - Dyn. SQL.: Ein mini REBIND kann nicht verhindert werden
    - Static SQL: SQL dyn. nachbauen & EXPLAIN verwenden
- Ergebnis interpretieren (nächste Folie)



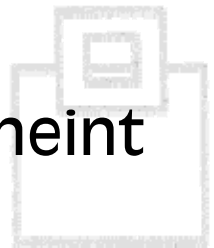
SOFTWARE ENGINEERING

# EXPLAINS Ausgabe interpretieren

---

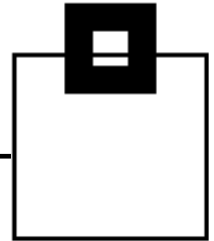


- Gleicher Zugriffspfad (PROC MS/MU evtl. verändert)
  - Statistiken wertlos
  - Veränderte Kostenschätzungen evtl. interessant
- Veränderter Zugriffspfad auf volatilem (leeren) Object
  - Statistiken nicht (direkt) verwenden, wenn Objekte
    - sich deutlich verkleinern
    - Mass Deletes aufweisen (ggfs. auch LOAD REPLACE)
- Veränderter Zugriffspfad (sonst)
  - sollte verwendet werden, auch wenn er schlechter scheint

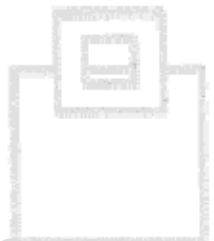


# Agenda

---



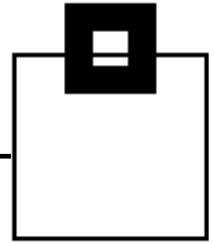
- Column distribution stats: Nahrung für den Optimizer
- Kosten-Nutzen Abwägung für RUNSTATS
- Vermeidung überflüssiger (Verteilungs-)Statistiken
- Ein praktikabler Ansatz auf Basis von COLGROUP bzw. FREQVAL
- Den Einfluss der Statistiken auf JOIN-Zugriffspfade prüfen
- **Zusammenfassung & Ausblick**
- **Fragen & Antworten**



SOFTWARE ENGINEERING

# Zusammenfassung & Ausblick

---



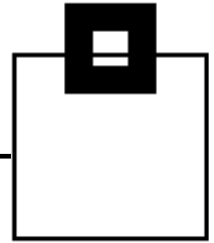
- Optimizer ohne erweiterte Statistiken durch korrelierte Daten verwirrt
- Allzuviel ist ungesund, auch bei RUNSTATS
- Die Auswirkungen von RUNSTATS und REBINDS einschätzen
- Wartung der (erweiterten) Statistiken nicht automatisch
- Finden Sie Ihre Methode
  - Kandidaten zu wählen und zu prüfen
  - Statistiken zu erstellen
  - Objektzustände zu prüfen/warten/verwalten



SOFTWARE ENGINEERING

# Fragen, Antworten & Spannung

---



Vielen Dank für Ihre Aufmerksamkeit! Dies ist einer der möglichen Zeitpunkte, um Fragen zu stellen.



Einige Details werden sich aber morgen ab 11:15 Uhr im Vortrag meines Kollegen Roy Boxwell klären.



Are you a Runstats Master?



SOFTWARE ENGINEERING