

# Willkommen zum DOAG Regionaltreffen

Index Rebuilds in großen  
hoch verfügbaren Oracle  
DBs

Rainer Klomps  
Trivadis GmbH

DOAG Regio Rhein-Main  
2014

BASEL    BERN    LAUSANNE    ZÜRICH    DÜSSELDORF    FRANKFURT A.M.    FREIBURG I.BR.    HAMBURG    MÜNCHEN    STUTTGART    WIEN



2013 © Trivadis

**trivadis**  
makes IT easier. ■ ■ ■

## About me ..

- Consultant , Trivadis Düsseldorf
- Arbeit mit Oracle seit 1996
  - Entwicklung (Forms, PL/SQL)
  - Support
  - Datenbankadministration
- Bei Trivadis seit 2012
- Schwerpunkt
  - Datenbankadministration



# AGENDA

1. Einführung
2. Warum liegt die Betonung auf „groß“ / „Hochverfügbarkeit“?
3. Ein Eindruck vom Wachstum des Index-Baums
4. Index Rebuild – Das Ziel
5. Index Rebuild – Indexzugriff
6. Index Rebuild – Index Range Scan
7. Hilfsmittel zur Analyse der Indexknoten
8. Und nicht vergessen ...
9. Ergebnisse / Beispiel
10. Fazit
11. Referenzen

# Einführung

- Angenommen, Sie haben Kopfschmerzen. Ist es gut Tabletten zu nehmen?
  - Ja, weil Sie dadurch die Kopfschmerzen bekämpfen können
  - Vielleicht, einige Menschen reagieren nicht auf die Tabletten
  - Es hängt davon ab ... Die Tabletten könnten helfen, aber schaden den Organen wie z.B. Magen und/oder Nieren
  - Nein, die Nebenwirkungen könnten schlimmer sein, als Kopfschmerzen.
- Angenommen, Sie haben Platz-, Performance und/oder I/O Probleme in Ihrer Datenbank. Ist es eine gute Idee die Indizes zu reorganisieren?
  - ...

# AGENDA

1. Einführung
2. Warum liegt die Betonung auf „groß“ / „Hochverfügbarkeit“?
3. Ein Eindruck vom Wachstum des Index-Baums
4. Index Rebuild – Das Ziel
5. Index Rebuild – Indexzugriff
6. Index Rebuild – Index Range Scan
7. Hilfsmittel zur Analyse der Indexknoten
8. Und nicht vergessen ...
9. Ergebnisse / Beispiel
10. Fazit
11. Referenzen

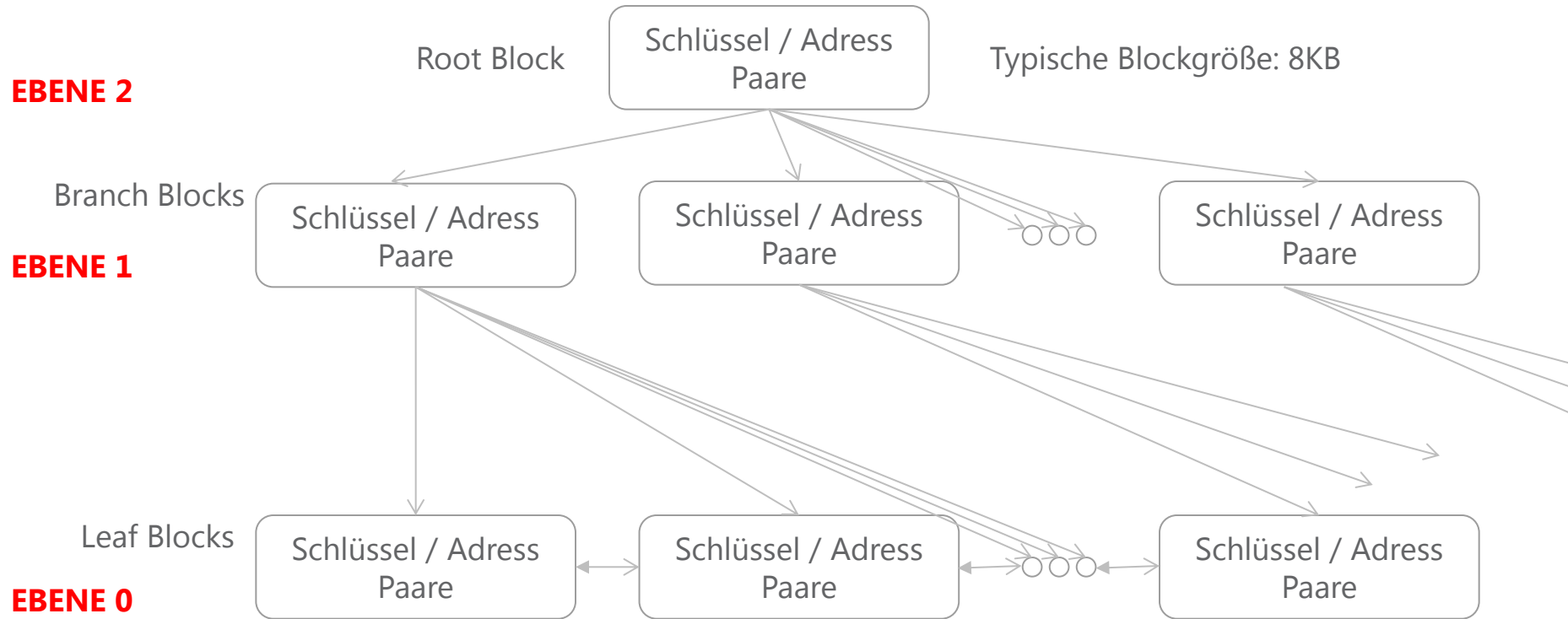
# Warum liegt die Betonung auf „groß“ / „Hochverfügbarkeit“?

- Große Hochverfügbarkeits-DBs sind sensibel bezüglich Ressourcen wie z.B.
  - CPU Nutzung
  - I/O Last
  - Locks
- Index Reorganisationen für große Indizes belasten die CPU und I/O Ressourcen
- Fehlende oder zu kleine Wartungsfenster
- In kleinen DBs ohne Hochverfügbarkeitsanforderung lassen sich oft Wartungsfenster finden, in denen sämtliche Indizes der DB reorganisiert werden können, ohne dass man sich um deren Notwendigkeit kümmert.
- Moderne Computer könnten so performant sein, dass man sich um möglicherweise nicht performante Indizes keine Gedanken machen muss.

# AGENDA

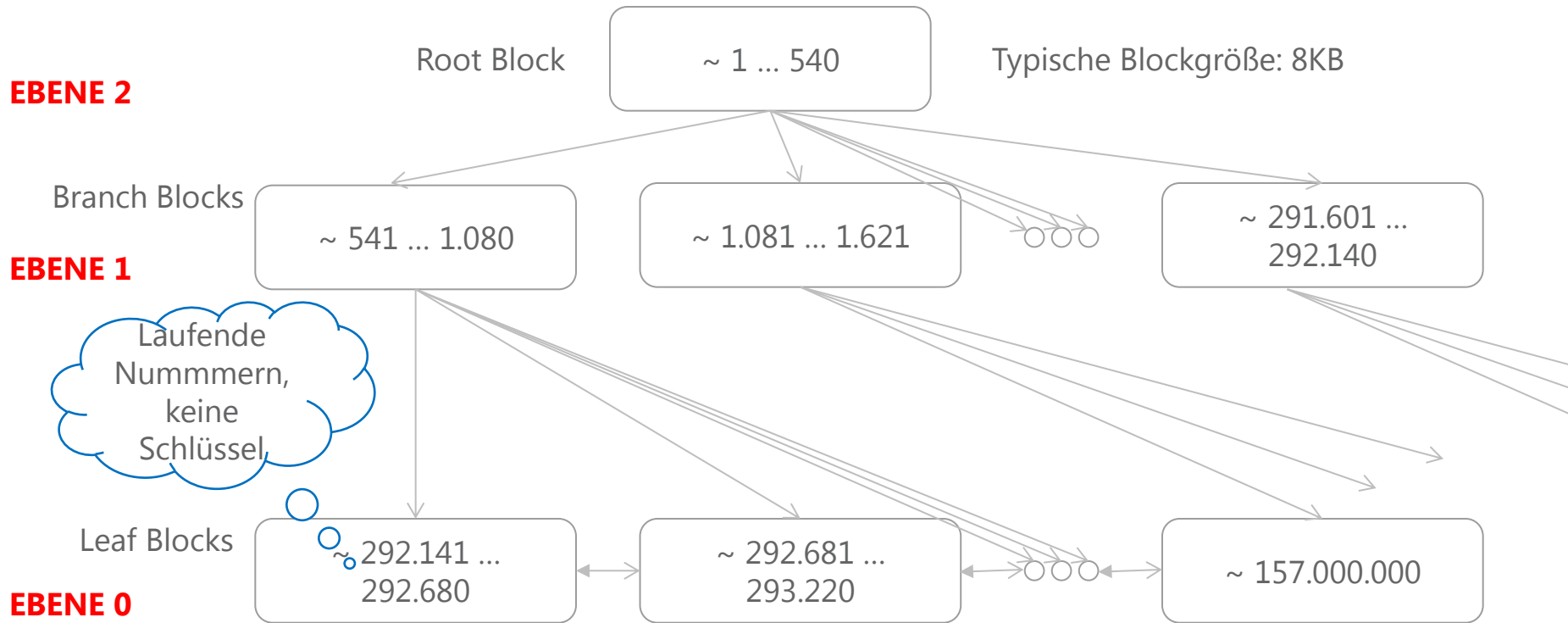
1. Einführung
2. Warum liegt die Betonung auf „groß“ / „Hochverfügbarkeit“?
3. Ein Eindruck vom Wachstum des Index-Baums
4. Index Rebuild – Das Ziel
5. Index Rebuild – Indexzugriff
6. Index Rebuild – Index Range Scan
7. Hilfsmittel zur Analyse der Indexknoten
8. Und nicht vergessen ...
9. Ergebnisse / Beispiel
10. Fazit
11. Referenzen

# Ein Eindruck vom Wachstum des Index-Baums



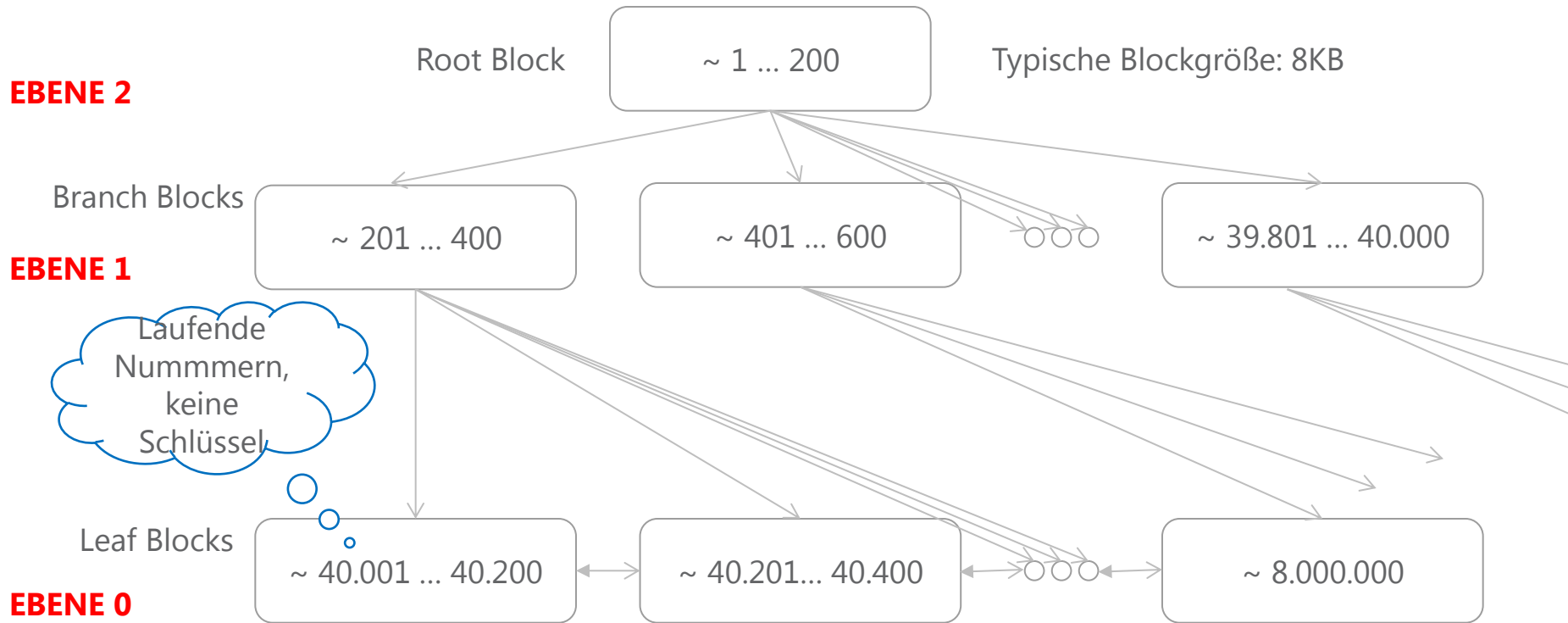


# Ein Eindruck vom Wachstum des Index-Baums



<Nummer> bezieht sich hier auf eine Spalte des Datentyps **number**

# Ein Eindruck vom Wachstum des Index-Baums

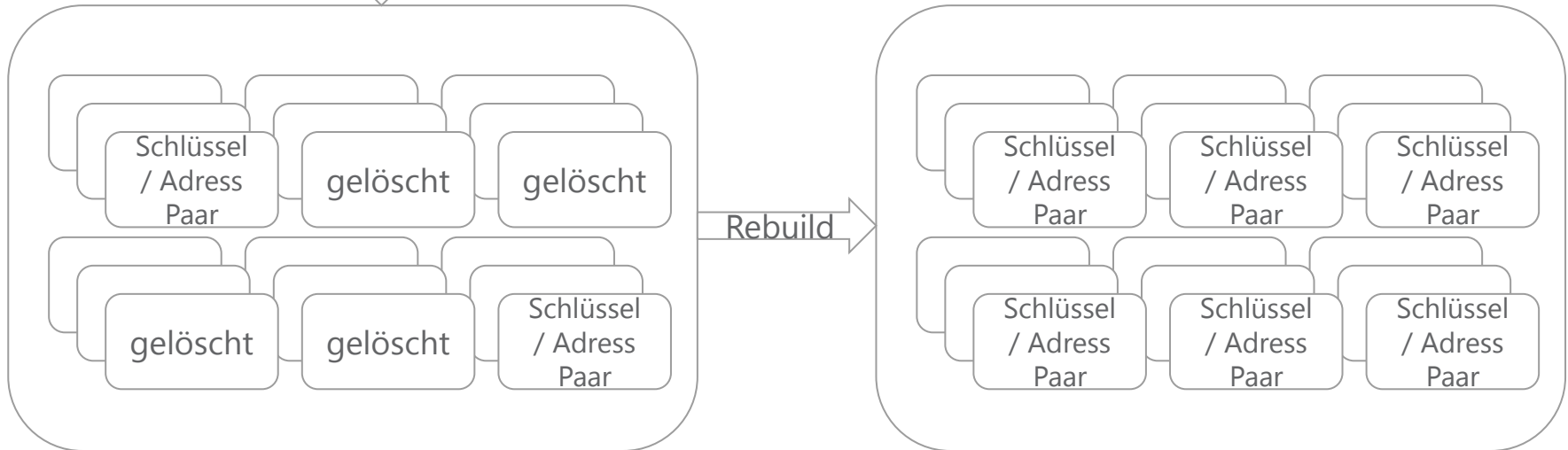
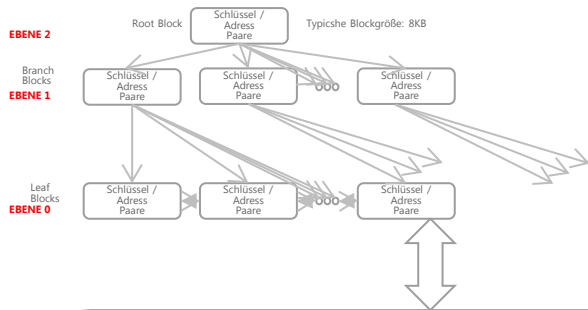


<Nummer> bezieht sich hier auf eine Spalte des Datentyps **varchar2(32)**, von denen jede einen 32 Zeichen Textstring enthält.

# AGENDA

1. Einführung
2. Warum liegt die Betonung auf „groß“ / „Hochverfügbarkeit“?
3. Ein Eindruck vom Wachstum des Index-Baums
4. **Index Rebuild – Das Ziel**
5. Index Rebuild – Indexzugriff
6. Index Rebuild – Index Range Scan
7. Hilfsmittel zur Analyse der Indexknoten
8. Und nicht vergessen ...
9. Ergebnisse / Beispiel
10. Fazit
11. Referenzen

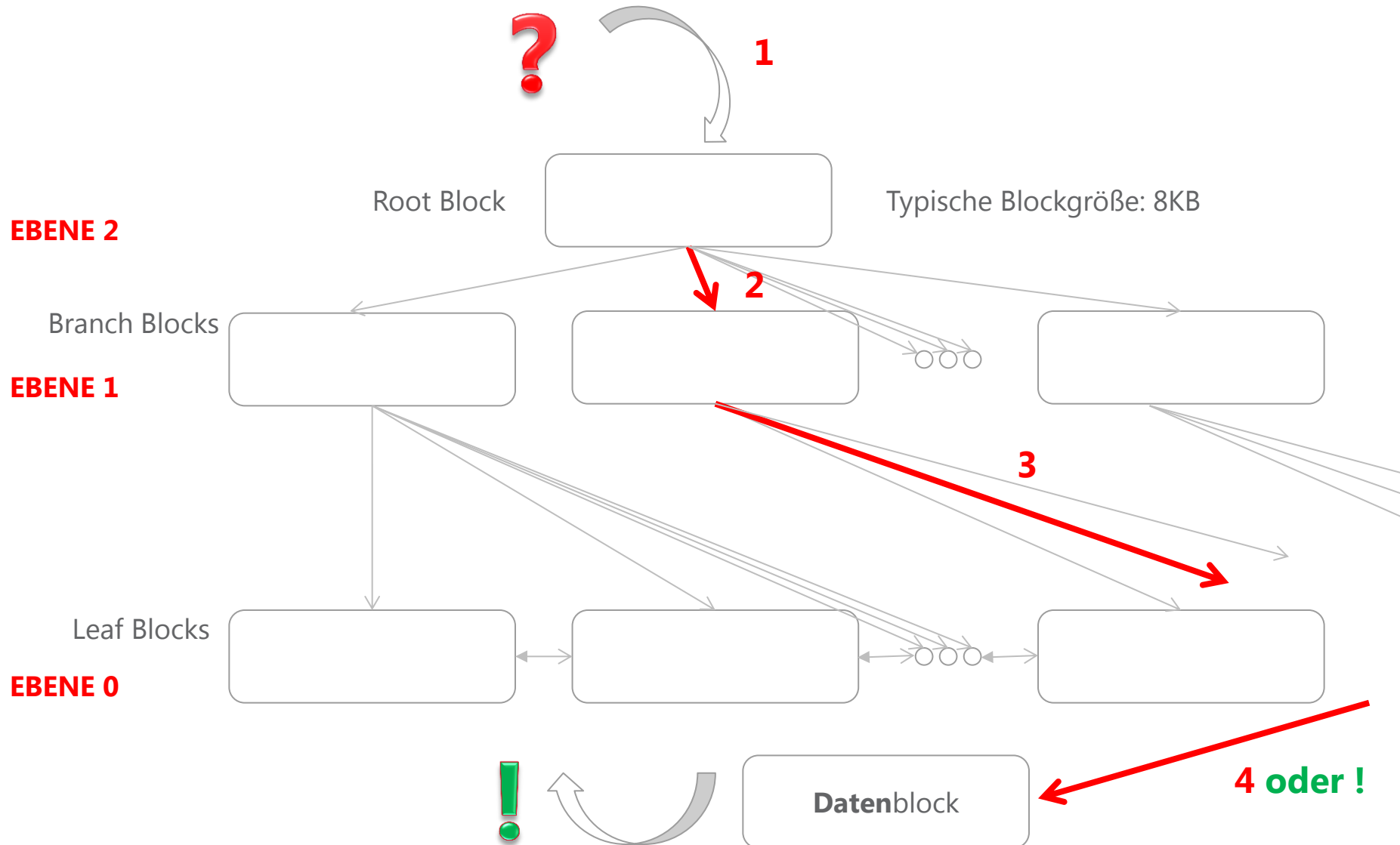
# Index Rebuild – Das Ziel



# AGENDA

1. Einführung
2. Warum liegt die Betonung auf „groß“ / „Hochverfügbarkeit“?
3. Ein Eindruck vom Wachstum des Index-Baums
4. Index Rebuild – Das Ziel
5. Index Rebuild – Indexzugriff
6. Index Rebuild – Index Range Scan
7. Hilfsmittel zur Analyse der Indexknoten
8. Und nicht vergessen ...
9. Ergebnisse / Beispiel
10. Fazit
11. Referenzen



# Index Rebuild – Indexzugriff



# Index Rebuild – Indexzugriff

Basierend auf dem Wachstum der Anzahl von Schlüsseln von Ebene zu Ebene, muss man höchstens drei oder vier Blöcke lesen, um die gewünschten Daten zu erhalten.

- Fazit für dieses Szenario:

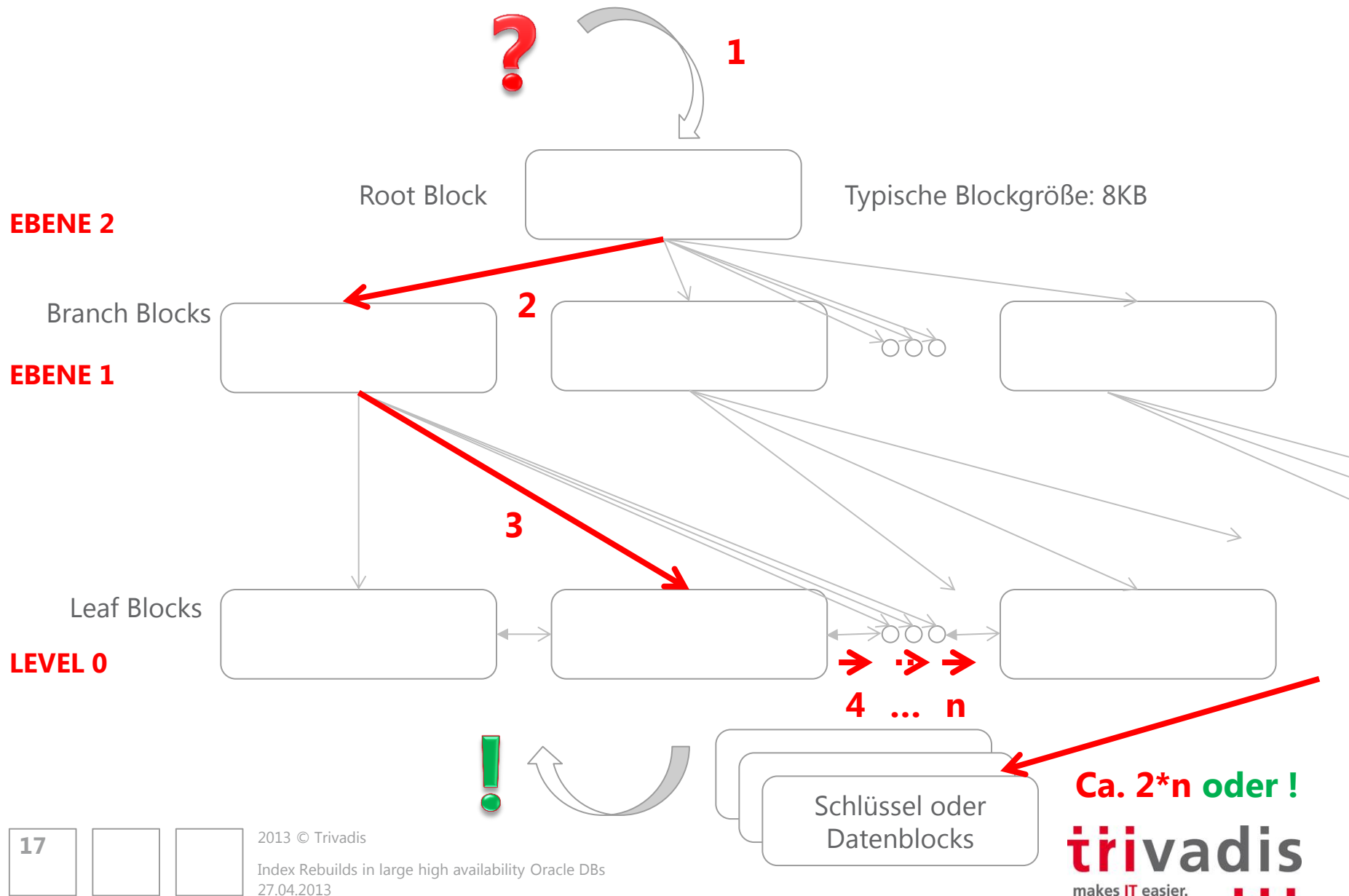
- Es ist fast niemals notwendig, Indizes zu reorganisieren, wenn Datenzugriffe über den gewöhnlichen Indexzugriff erfolgen. 
- Man kann die Performance bei einem Indexzugriff kaum merklich durch Reorganisation der Indizes steigern 

# AGENDA

1. Einführung
2. Warum liegt die Betonung auf „groß“ / „Hochverfügbarkeit“?
3. Ein Eindruck vom Wachstum des Index-Baums
4. Index Rebuild – Das Ziel
5. Index Rebuild – Indexzugriff
6. Index Rebuild – Index Range Scan
7. Hilfsmittel zur Analyse der Indexknoten
8. Und nicht vergessen ...
9. Ergebnisse / Beispiel
10. Fazit
11. Referenzen



# Index Rebuild – Index Range Scan



# Index Rebuild – Index Range Scan

Falls Abfragen viele Index Range Scans benutzen, könnten hierbei sehr viele spärlich besetzte Blattknoten des Indexbaums durchlaufen werden.

## ■ Fazit:

- Es könnte sich lohnen, Indizes zu reorganisieren, um leere bzw. spärlich besetzte Blattknoten zu eliminieren.
- Indexreorganisation wird die Performance der DB erhöhen, falls der Index viele Leereinträge enthält.
- Falls man immer wieder eine sehr große Anzahl von Datensätzen entfernt und wieder einfügt, könnte die Reorganisation teurer werden, als die Indizes unverändert zu lassen.



# AGENDA

1. Einführung
2. Warum liegt die Betonung auf „groß“ / „Hochverfügbarkeit“?
3. Ein Eindruck vom Wachstum des Index-Baums
4. Index Rebuild – Das Ziel
5. Index Rebuild – Indexzugriff
6. Index Rebuild – Index Range Scan
7. Hilfsmittel zur Analyse der Indexknoten
8. Und nicht vergessen ...
9. Ergebnisse / Beispiel
10. Fazit
11. Referenzen

# Hilfsmittel zur Analyse der Indexknoten

Scannen der Blattknoten mittels sys\_op\_lbid

```
SQL > break on report skip 1
```

```
SQL > compute sum of blocks on report
```

```
SQL > column ind_id new_value candidate_idx
```

```
SQL > select object_id ind_id from user_objects  
       where object_name = '&MY_INDEX';
```

```
SQL > select rows_per_block, count(*) blocks from  
       ( select sys_op_lbid( &&candidate_idx , 'L', &&MY_TABLE..rowid)  
         as block_id, count(*) as rows_per_block from &&MY_TABLE  
         where A is not null  
         group by sys_op_lbid(&&candidate_idx , 'L', &&MY_TABLE..rowid)  
       ) group by rows_per_block order by rows_per_block;
```

# Hilfsmittel zur Analyse der Indexknoten

Output basierend auf einer Tabelle mit 1.000.000 Zeilen, indiziert auf einer numerischen Spalte die die Werte 1 ... 1.000.000 enthält.

ROWS_PER_BLOCK	BLOCKS
486	1
533	1856
552	1
571	16
578	1
<b>Summe</b>	<b>1875</b>

# Hilfsmittel zur Analyse der Indexknoten

```
SQL > ALTER SESSION SET EVENTS
```

```
'IMMEDIATE TRACE NAME TREEDUMP LEVEL &OBJECT_ID_IHRES_INDEX';
```

- Wie üblich bei dem 'set events' Befehl, erfolgt die Ausgabe der Trace-Informationen in eine Datei im Alertlog Verzeichnis.

# Hilfsmittel zur Analyse der Indexknoten

- Inhalt der generierten Trace-Datei:
  - Index auf einer numerischen Spalte initial mit den Werten 1 ... 1.000 geladen
  - Anschließend alle Zeilen mit geraden Nummern gelöscht

```
----- begin tree dump
```

```
branch: 0x8027d13 134380819 (0: nrow: 3, level: 1)
```

```
  leaf: 0x8027d14 134380820 (-1: nrow: 485 rrow: 243) → 242 gelöscht
```

```
  leaf: 0x8027d15 134380821 (0: nrow: 479 rrow: 239) → 240 gelöscht
```

```
  leaf: 0x8027d16 134380822 (1: nrow: 36 rrow: 18) → 18 gelöscht
```

```
----- end tree dump
```

# Hilfsmittel zur Analyse der Indexknoten

- View "INDEX\_STATS"
  - HEIGHT
  - BLOCKS, LF\_BLKs, DEL\_LF\_BLKs, BR\_BLKs, DEL\_BR\_BLKs
  - BR\_ROWS, LF\_ROWS, DEL\_BR\_ROWS, DEL\_LF\_ROWS
- Erfordert Aufruf von ...

```
ANALYZE INDEX <index_name> VALIDATE STRUCTURE;
```

- In INDEX\_STATS ist immer höchstens ein Index sichtbar
- Statement führt zu einer Sperre des Index





# Hilfsmittel zur Analyse der Indexknoten

- Tabellen / Package von MOS 989186.1 zur Bestimmung der benutzten und Berechnung der minimal benötigten Index Blöcke (“Script to investigate a b-tree index structure”)
- Erfordert aktuelle Index Statistiken (➔ number of rows)
- Schemaspezifische Indexblock Berechnung basierend auf
  - Anzahl der Zeilen
  - Blockgröße
  - Datentyp der Spalten
  - Indexknoten Overhead
  - Eindeutigkeit / Mehrdeutigkeit des Index
- Ohne Sperren



# Hilfsmittel zur Analyse der Indexknoten

- Basierend auf sys\_op\_lbid (s.o.)
- Geringfügige Änderungen gegenüber MOS 989186.1:
  - Hinzufügen einer virtuellen Spalte die das Verhältnis von allokierten Blöcken zu minimal benötigten Blöcken angibt
  - Achtung: Spalte idx\_layout unterschiedlich deklariert in der Tabelle INDEX\_LOG (CLOB) und INDEX\_HIST (VARCHAR2(4000))

# Hilfsmittel zur Analyse der Indexknoten

- Package INDEX\_UTIL
  - Prozedur INSPECT\_INDEX
  - Prozedur INSPECT\_SCHEMA
- Tabellen INDEX\_LOG, INDEX\_HIST

OWNER	NOT NULL VARCHAR2 (30)
INDEX_NAME	NOT NULL VARCHAR2 (30)
LAST_INSPECTED	DATE
LEAF_BLOCKS	NUMBER
TARGET_SIZE	NUMBER
IDX_LAYOUT	<b>CLOB</b>

# Hilfsmittel zur Analyse der Indexknoten

```
alter table index_log  
  add relationship  
  as (round(LEAF_BLOCKS/TARGET_SIZE,2)) virtual;
```

```
alter table index_hist  
  add relationship  
  as (round(LEAF_BLOCKS/TARGET_SIZE,2)) virtual;
```

# AGENDA

1. Einführung
2. Warum liegt die Betonung auf „groß“ / „Hochverfügbarkeit“?
3. Ein Eindruck vom Wachstum des Index-Baums
4. Index Rebuild – Das Ziel
5. Index Rebuild – Indexzugriff
6. Index Rebuild – Index Range Scan
7. Hilfsmittel zur Analyse der Indexknoten
8. **Und nicht vergessen ...**
9. Ergebnisse / Beispiel
10. Fazit
11. Referenzen

## Und nicht vergessen ...

- ... Reorganisieren Sie Ihren Index online  
(zur Nutzung von Online Operationen wird die Enterprise Edition benötigt, kein locking)

```
alter index <index_name> rebuild online;
```

- Vielleicht auch interessant für Sie
  - Index Reorganisation nutzt "INDEX FAST FULL SCAN"
  - Index Reorganisation **online** nutzt "TABLE ACCESS FULL"

## Und nicht vergessen ...

- ... Parallelverarbeitung **AUSPROBIEREN**, falls Sie genügend Ressourcen haben
  - (zur Nutzung von Parallelverarbeitung wird die Enterprise Edition benötigt)

```
alter session force parallel ddl parallel <n>;
```

```
alter index <index_name> rebuild online parallel <n>;
```

- ... Zurückschalten auf den gewünschten Parallelitätsgrad !

```
alter index <index_name> noparallel;
```

```
alter index <index_name> parallel; --> default degree
```

```
alter index <index_name> parallel <m>;
```

## Und nicht vergessen ...

- ... coalescing hilft oft auch schon, ist erheblich schneller und benötigt weniger Ressourcen, allerdings ist das Ergebnis nicht so gut, wie das einer Reorganisation.

```
alter index <index_name> coalesce;
```

- Daumenregel, rebuild/coalesce Indizes, falls sie mehr als 20% gelöschte Zeilen enthalten.



## Und nicht vergessen ...

- ... Nutzung dedizierter Tablespaces mit größeren Datenblöcken (32 KB) für extrem große Indizes kann ebenfalls hilfreich sein.

# AGENDA

1. Einführung
2. Warum liegt die Betonung auf „groß“ / „Hochverfügbarkeit“?
3. Ein Eindruck vom Wachstum des Index-Baums
4. Index Rebuild – Das Ziel
5. Index Rebuild – Indexzugriff
6. Index Rebuild – Index Range Scan
7. Hilfsmittel zur Analyse der Indexknoten
8. Und nicht vergessen ...
9. Ergebnisse / Beispiel
10. Fazit
11. Referenzen

# Ergebnisse / Beispiel

Lauf	Name	Höhe	Blöcke	Branch Blöcke	benötigter Speicher
Initial	ORIGINAL_INDEX	3	3.584	7	27.344.224
Initial	COALESCE_IDX	3	3.584	7	27.344.224
Initial	REBUILD_IDX	3	3.584	7	27.344.224
Löschen	ORIGINAL_INDEX	3	3.584	7	27.344.224
Löschen	COALESCE_IDX	3	3.584	7	27.344.224
Löschen	REBUILD_IDX	3	3.584	7	27.344.224
Coalesce	COALESCE_IDX	3	3.584	7	9.200.224
Rebuild	REBUILD_IDX	3	1280	3	9.144.096

# AGENDA

1. Einführung
2. Warum liegt die Betonung auf „groß“ / „Hochverfügbarkeit“?
3. Ein Eindruck vom Wachstum des Index-Baums
4. Index Rebuild – Das Ziel
5. Index Rebuild – Indexzugriff
6. Index Rebuild – Index Range Scan
7. Hilfsmittel zur Analyse der Indexknoten
8. Und nicht vergessen ...
9. Ergebnisse / Beispiel
10. Fazit
11. Referenzen

# Fazit

- Angenommen, Sie haben Kopfschmerzen. Ist es gut Tabletten zu nehmen?
  - Ja, weil Sie dadurch die Kopfschmerzen bekämpfen können
  - Vielleicht, einige Menschen reagieren nicht auf die Tabletten
  - Es hängt davon ab ... Die Tabletten könnten helfen, aber schaden den Organen wie z.B. Magen und/oder Nieren
  - Nein, die Nebenwirkungen könnten schlimmer sein, als Kopfschmerzen.
- Angenommen, Sie haben Platz-, Performance und/oder I/O Probleme in Ihrer Datenbank. Ist es eine gute Idee die Indizes zu reorganisieren?
  - ...

# AGENDA

1. Einführung
2. Warum liegt die Betonung auf „groß“ / „Hochverfügbarkeit“?
3. Ein Eindruck vom Wachstum des Index-Baums
4. Index Rebuild – Das Ziel
5. Index Rebuild – Indexzugriff
6. Index Rebuild – Index Range Scan
7. Hilfsmittel zur Analyse der Indexknoten
8. Und nicht vergessen ...
9. Ergebnisse / Beispiel
10. Fazit
11. Referenzen

# Referenzen

- Richard Foote:
  - Oracle B-Tree Indexes Internals: Rebuilding the Truth
  - Feb. 12th 2007
  - [http://www.dbafan.com/book/oracle\\_index\\_internals.pdf](http://www.dbafan.com/book/oracle_index_internals.pdf)
  - (Emphasize on index related myths)
- Richard Foote:
  - So When Does An Oracle B-Tree Index Increase In Height ? (Almost Grown)
  - Apr. 3rd 2008
  - <http://richardfoote.wordpress.com/2008/04/03/so-when-does-an-oracle-b-tree-index-increase-in-height-almost-grown/>

# Referenzen

- MOS Note 989186.1
  - Script to investigate a b-tree index structure
  - ~~Mar, 1st 2013~~ Apr. 7<sup>th</sup> 2014
- Daniel A. Morgan
  - sys\_op\_lbid
  - Aug. 20th 2012
  - [http://www.morganslibrary.org/reference/sys\\_op\\_lbid.html](http://www.morganslibrary.org/reference/sys_op_lbid.html)
  - (Anmerkung RKL: LBID steht für **Leaf Block ID**)



Vielen Dank für  
Ihre  
Aufmerksamkeit.

Trivadis GmbH  
Rainer Klomps

Werdener Straße 4  
40227 Düsseldorf  
Tel. +49 211 58 66 64 70  
Fax +49 211 58 66 64 71

info@trivadis.com  
www.trivadis.com

BASEL    BERN    LAUSANNE    ZÜRICH    DÜSSELDORF    FRANKFURT A.M.    FREIBURG I.BR.    HAMBURG    MÜNCHEN    STUTTGART    WIEN



2013 © Trivadis

**trivadis**  
makes IT easier. ■ ■ ■