

Oracle Advanced Compression 10g versus 11g

Platz in der Datenbank optimal nützen

Ihr Partner für Schulung, Betreuung und Beratung rund um die **Oracle-Datenbank**

Gunther Pippèr

Über uns

- Oracle Consulting seit Oracle 7.1
- IT Service Provider und Remote-Backup-Dienstleister
 - Eigenes Rechenzentrum
 - Wir sichern Ihre Datenbank per Remote Backup



- Eigenes Java-Framework XCP

Agenda

- Oracle Compression for Relational Data
 - 10g – Table und Index Compression
 - 11g – OLTP Table Compression
 - Deduplizierung auf Lob's
- Was bringt das Feature
- Wer kann es überhaupt nützen

Szenario

Datenbank wachen immer mehr – Täglich –
Stündlich

Wie mit den Datenmengen umgehen?

Wie den Datenmengen
Herr werden?

Die Problemstellung

- Wie lassen sich Datenmengen verkleinern
- Komprimierung
 - Datenmenge reduzieren, indem eine günstigere Repräsentation bestimmt wird, mit der sich die gleichen Informationen in kürzerer Form darstellen lassen
- Deduplizierung
 - Datenmenge reduzieren durch einen Prozess, der redundante Daten identifiziert und eliminiert



Das Ziel

- Platzbedarf auf den Festplatten reduzieren
- Weniger I/O beim Lesen
- Speicher effizienter ausnützen

Was lässt sich komprimieren

- Tabellen
 - Oracle9i und 10g Release2 Enterprise Edition
 - nur in speziellen DML Befehlen – Bulk Load
 - 11g – alle DML Operationen
- Materialised Views
 - Ab 9i bis heute
- B-tree Index und Index organized tables
 - Ab 8i bis heute
- Deduplizierung von LOB's
 - 11g

Welche DB Lizenzierung

- Compression Basis => Enterprise Edition
 - Advanced Compression => Oracle 11g Option

Feature/Option	SE1	SE	EE	Notes
Oracle Advanced Compression	N	N	Y	Extra cost option
Direct-Load Table Compression	N	N	Y	

– 9i

- Metalink Doc ID: 269040.1
Differences Between Enterprise, Standard and Personal Editions on Oracle 9.2

– 10g

- Metalink Doc ID: 465465.1
Differences Between Enterprise, Standard and Personal Editions on Oracle 10.2

– 11g

- Metalink Doc ID: 465465.1
Differences Between Enterprise, Standard and Personal Editions on Oracle 11.1

Table Compression einsetzen

- Überblick
- Tabellen

```
Create table TT (id number(9), texte varchar2(256))  
compress  
for all operations;
```

11g – Advanced

11g

```
Select table_name, compression, compress_for FROM  
USER_TABLES;
```

TABLE_NAME	COMPRESSION	COMPRESS_FOR
TT	ENABLED	FOR ALL OPERATIONS
EMP	DISABLED	

Überprüfen mit?

Table Compression

- Auf Tablespace Ebene

Kann auch auf Tablespace Ebene definiert werden!

Alle Objekte erben dann diese Eigenschaft.

```
CREATE TABLESPACE comp_test  
  
default COMPRESS FOR ALL OPERATIONS  
  
DATAFILE 'C:\oracle\oradata\GPI\comp_test.DBF'  
SIZE 100M AUTOEXTEND ON NEXT 2M MAXSIZE 1000M  
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K  
SEGMENT SPACE MANAGEMENT AUTO;
```

Table Compression in 10g einsetzen

- Funktioniert nur mit:
 - Direct path SQL*Loader
 - Serial INSERT with an APPEND hint
 - Parallel INSERT
 - CREATE TABLE ... AS SELECT



Bestehende Tabellen komprimieren

- Tabelle neu aufbauen
 - Move Compress

```
ALTER TABLE TT MOVE COMPRESS;
```



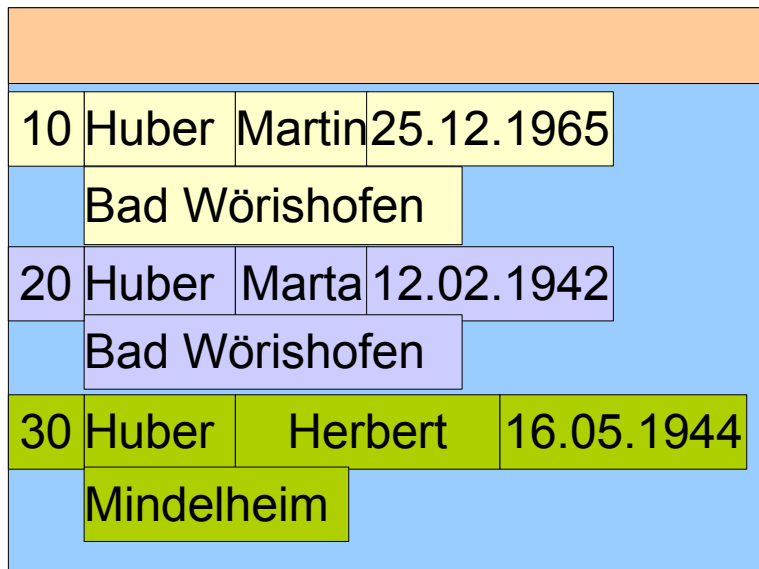
Prinzip – Immer pro Block!

- Beispiel

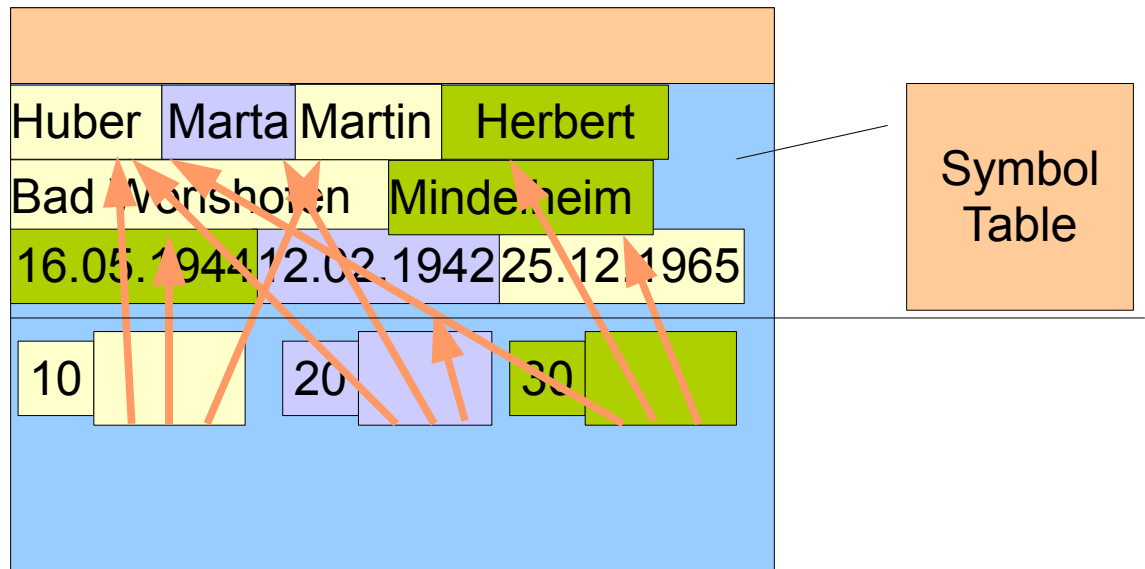
ID	Name	Vorname	Gebt.	Ort
10	Huber	Martin	25.12.1965	Bad Wörishofen
20	Huber	Marta	12.02.1942	Bad Wörishofen
30	Huber	Herbert	16.05.1944	Mindelheim



Block normal



Block komprimiert



Ein Blick auf die Blöcke

- Block Dump erzeugen mit

```
Select  dbms_rowid.rowid_relative_fno(rowid)  fn#  
        , dbms_rowid.rowid_block_number(rowid)  blk#  
        , dbms_rowid.rowid_row_number(rowid)    brow#  
from com_t where object_id = '119';
```

FN#	BLK#	ROW#
6	200364	115

```
alter system dump datafile 6 block 200364;
```

Tracefile suchen

SQL> show parameter user_dump_dest



Block Dump komprimiert

```
tab 1, row 115, @0x143f
tl: 22 fb: --H-FL-- lb: 0x0 cc: 15
col 0: *NULL*
col 1: [ 5] 56 41 4c 49 44
col 2: [ 1] 4e
col 3: [ 1] 4e
col 4: [ 1] 4e
col 5: [ 3] 53 59 53
col 6: *NULL*
col 7: [ 2] c1 02
col 8: [ 5] 54 41 42 4c 45
col 9: [ 7] 78 6b 0a 0f 0b 0a 0b
col 10: [19] 32 30 30 37 2d 31 30 2d 31 35 3a 31 30 3a 30 39
3a 31 30
col 11: [ 7] 78 6b 0a 0f 0b 0a 0b
col 12: [ 3] c2 02 14
col 13: [ 8] 44 45 46 52 4f 4c 45 24
col 14: [ 3] c2 02 14
bindmp: 2c 00 04 08 03 cb c2 02 14 d0 44 45 46 52 4f 4c 45 24
cb c2 02 14
```

Verweis Symboltable

Block Dump unkomprimiert

```
tab 0, row 27, @0x1672
tl: 82 fb: --H-FL-- lb: 0x0 cc: 14
col 0: [ 3] 53 59 53
col 1: [ 8] 44 45 46 52 4f 4c 45 24
col 2: *NULL*
col 3: [ 3] c2 02 14
col 4: [ 3] c2 02 14
col 5: [ 5] 54 41 42 4c 45
col 6: [ 7] 78 6b 0a 0f 0b 0a 0b
col 7: [ 7] 78 6b 0a 0f 0b 0a 0b
col 8: [19] 32 30 30 37 2d 31 30 2d 31 35 3a 31 30 3a 30 39
3a 31 30
col 9: [ 5] 56 41 4c 49 44
col 10: [ 1] 4e
col 11: [ 1] 4e
col 12: [ 1] 4e
col 13: [ 2] c1 02
```


Auswirkungen

- Je größer der Block, um so besser die Kompression
 - Kardinalität der Wert einer Tabelle entscheidend für die Qualität der Kompression
- => Daten sortiert einfügen beim Laden
- Daten im Speicher und auf Platte in gleicher Darstellung => Block Buffer wird besser ausgenützt
 - Geringer CPU Overhead

Praxis

- Beispiel:

```
-- unkomprimierte Tabelle  
create table uncom_t  
as select * from all_objects;
```

Abgelaufen: 00:00:19.03

```
-- komprimierte Tabelle  
create table com_t compress for all operations  
as select * from all_objects;
```

Abgelaufen: 00:00:23.79

Messen der Komprimierung

- Wie sehe ich den Effekt?

```
select count (distinct
dbms_rowid.rowid_block_number(rowid)) "TABLE BLOCK
COUNT" from uncom_t;
```

```
TABLE BLOCK COUNT
```

```
-----
1000
```

```
select count (distinct
dbms_rowid.rowid_block_number(rowid)) "TABLE BLOCK
COUNT" from com_t;
```

```
TABLE BLOCK COUNT
```

```
-----
310
```

Messen der Komprimierung

- Wie sehe ich den Effekt?

```
select avg(count(*)) "AVERAGE RECORDS PER BLOCK"  
from uncom_t group by  
dbms_rowid.rowid_block_number(rowid);
```

AVERAGE RECORDS PER BLOCK

67,962

```
select avg(count(*)) "AVERAGE RECORDS PER BLOCK"  
from com_t group by  
dbms_rowid.rowid_block_number(rowid);
```

AVERAGE RECORDS PER BLOCK

219,235484

Messen der Komprimierung

- Wie sehe ich den Effekt?

```
format column segment_name a20

select segment_name, round(bytes/1024/1024) "MB"
  from dba_segments
 where owner='TEST01'
       and segment_name in ('UNCOM_T', 'COM_T');
```

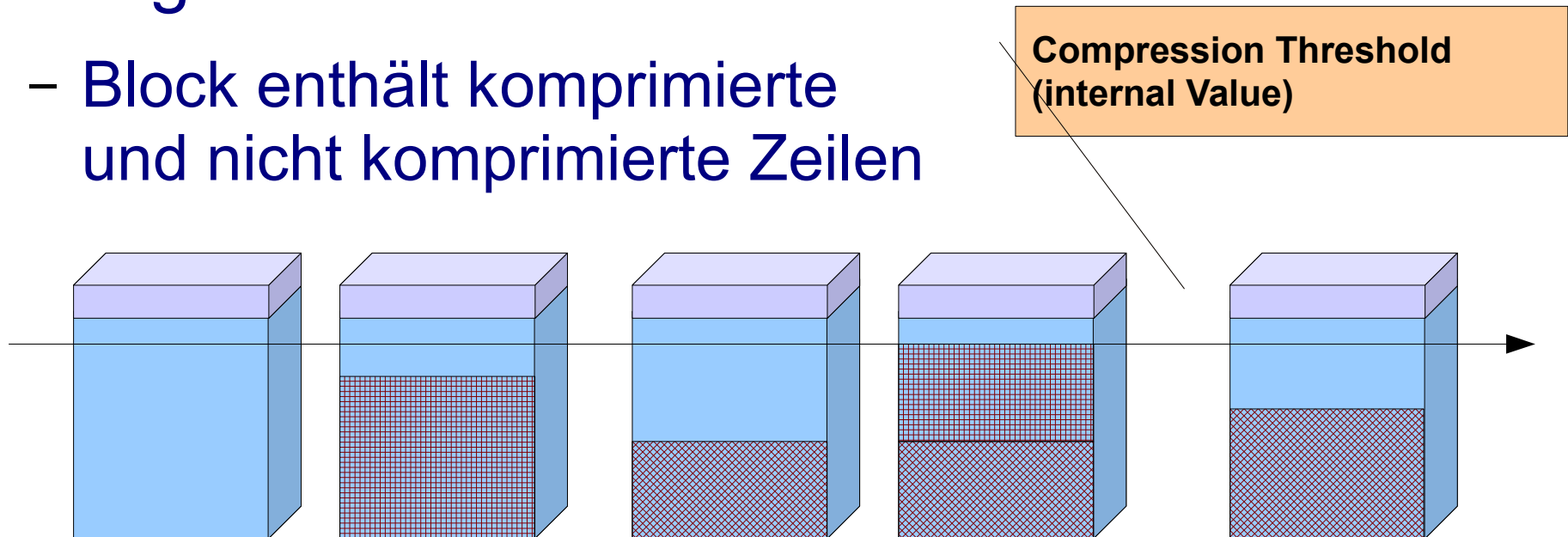
SEGMENT_NAME	MB
UNCOM_T	8
COM_T	3

Ab 11g Release 2

DBMS_TABCOMP.GET_RATIO

11g Block Level Batch Compression

- Wie gut funktioniert das?
 - Block enthält komprimierte und nicht komprimierte Zeilen



- Einzelne Insert und Updates verursachen keine Komprimierung
- Erst wenn Schwellwert erreicht wird, wird komprimiert

Test DML Operationen

- Kardinalität einer Spalte massiv ändern:

```
set timing on
```

```
update uncom_t set owner=owner||to_char(object_id);
```

```
67962 Zeilen wurden aktualisiert.  
Abgelaufen: 00:00:17.23
```

```
update com_t set owner=owner||to_char(object_id);
```

```
67963 Zeilen wurden aktualisiert.  
Abgelaufen: 00:02:24.69
```

Update bis zu 4-8 Mal langsamere



Test DML Operationen

- Datenvolumen pro Row vergrößern

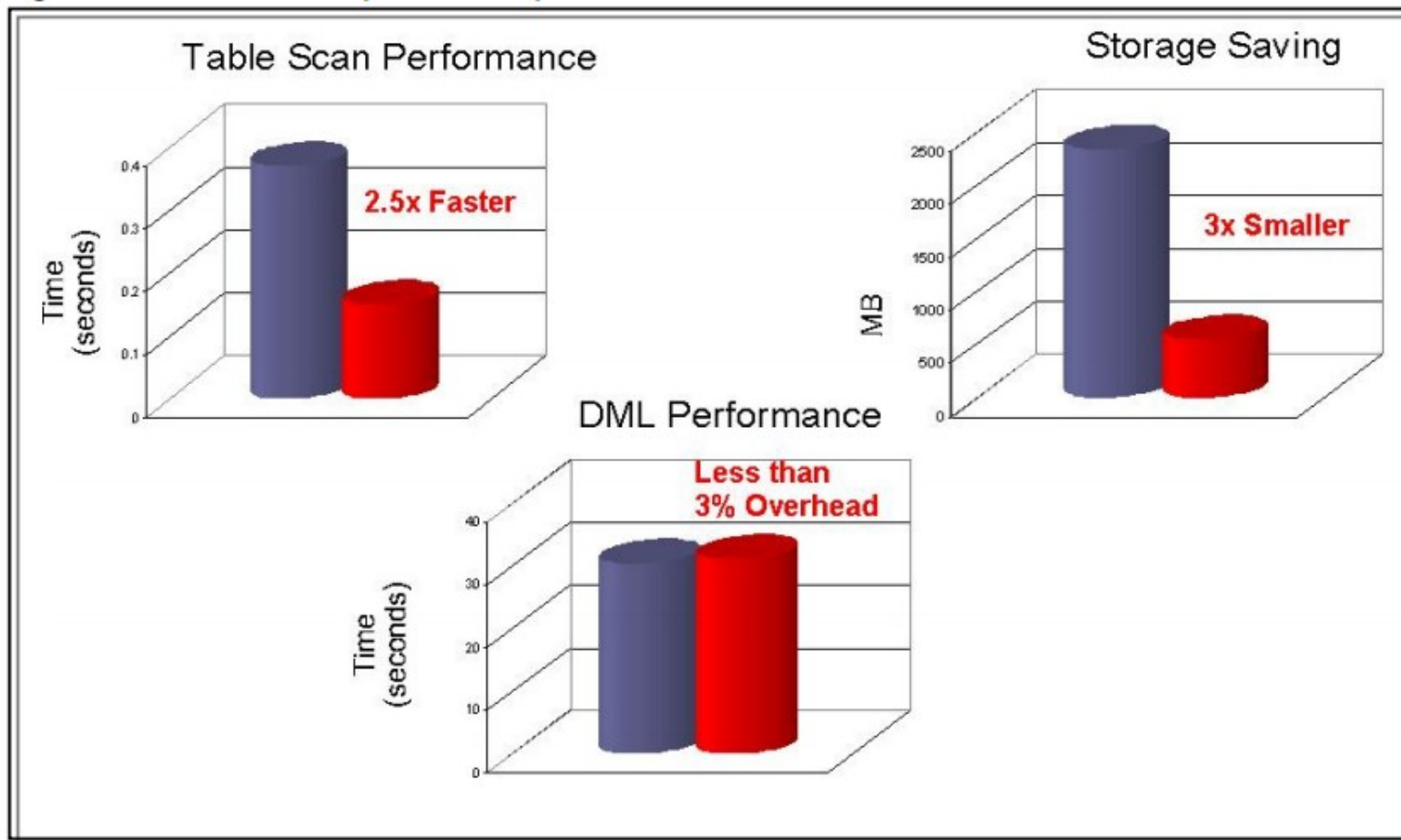
```
alter table com_t modify (SUBOBJECT_NAME  
varchar2(4000));  
  
update com_t set  
SUBOBJECT_NAME=rpadd(OBJECT_NAME, '4000', 'X');
```



10h später?
BUG?

Performance

- Das verspricht Oracle



Aus: Oracle Advanced Compression - An Oracle White Paper - April 2008

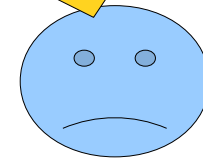
TPC-C Test

- Testwerkzeug: QUEST Benchmark Factory
- 1. Test – Anlegen von 1GB Testdaten
 - Unkomprimierter Tablespace
 - => Größe 1,703,419,904 Bytes Datei Größe
 - Komprimierter Tablespace
 - => Größe 1,565,532,160 Bytes Datei Größe

TPC-C Test

- 2. Test Laufzeit TPC-C TEST
 - Unkomprimierter Tablespace
 - Komprimierter Tablespace

Kein nennenswerter Unterschied?



Einschränkungen

- Table Compression Is Limited To 255 Columns
 - Metalink Doc ID: 443843.1
 - Oracle Server - Enterprise Edition - Version: 9.2.0.3 to 11.1.0.6
 - If a table has more than 255 columns, it does not get compressed. The statement executed to compress the table will execute successfully but data will not be compressed.

Einschränkungen

- Unable To Drop An Unused Column From A Compressed Table Due To ORA-12996
 - Metalink Doc ID: 558630.1
 - Oracle Server - Enterprise Edition - Version: 10.1.0.2 to 10.2.0.1

Selber komprimieren ab 10g

- Using the new UTL_COMPRESS
Oracle Supplied Package
 - Metalink Doc ID: 249974.1

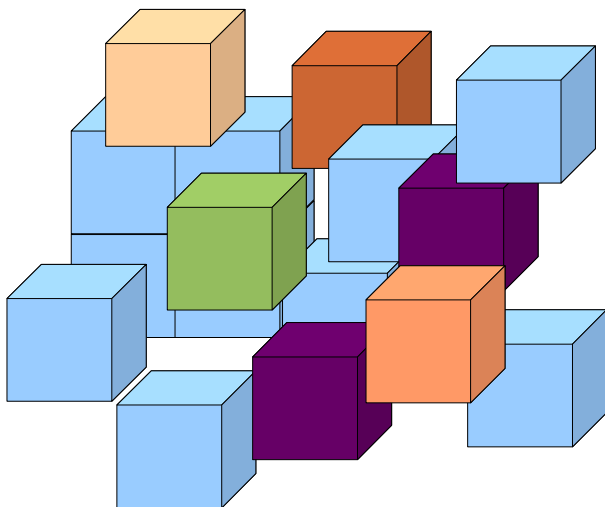
```
utl_compress.lz_compress(  
src      IN BLOB,  
quality IN BINARY_INTEGER DEFAULT 6)  
RETURN BLOB;
```

```
utl_compress.lz_uncompress(src IN BLOB) RETURN BLOB;
```

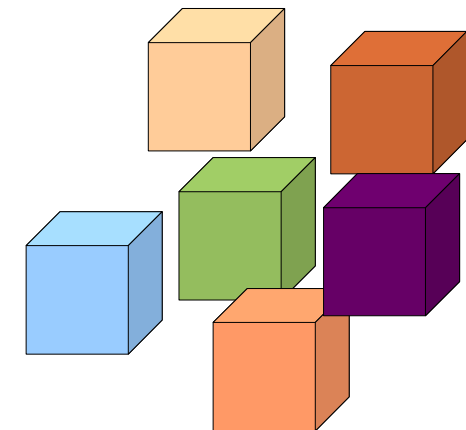
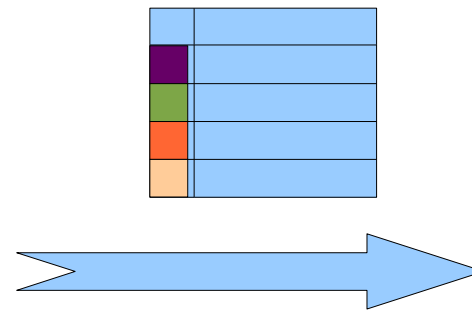
11g SecureFile LOB Deduplication

- Lobs werden nur einfach gespeichert
 - Ideal für Content Management wie ein E-Mail Archiv

Datenvolumen
brutto



Secure Hash pro Datei



Nur eine Version
Wird gespeichert!

Innerhalb einer Tabelle, Partition, Sub-Partition

SecureFiles Compression

- Lob komprimiert speichern
- Transparent
 - Random Read und write unterstützt
 - Automatische Erkennung ob eine Datei bereits komprimiert ist => Falls Ja, überspringen
 - Auto-turn off, Falls Komprimierung nichts bringt

SecureFiles Syntax

- Compression Syntax

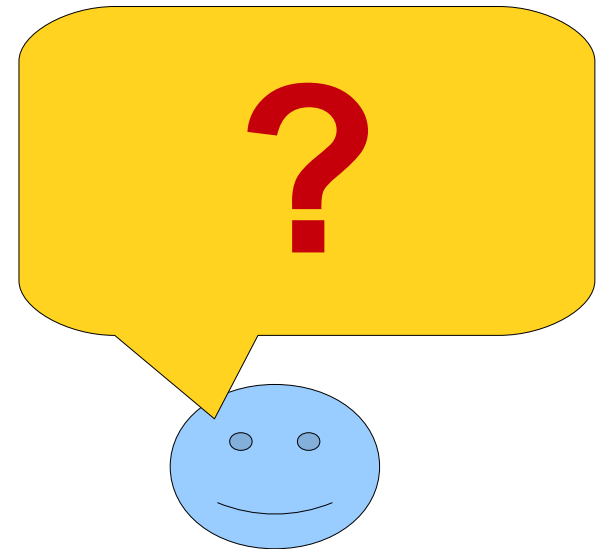
```
CREATE TABLE t1 ( a CLOB )  
  LOB(a) STORE AS SECUREFILE (  
    COMPRESS HIGH  
    CACHE  
  );
```

- Deduplication Syntax

```
CREATE TABLE t1 ( a CLOB )  
  LOB(a) STORE AS SECUREFILE (  
    DEDUPLICATE  
    CACHE  
  );
```

Ihre Erfahrung?

- Compression im Einsatz?
- Erfolg damit?



Zusammenfassung

- Advanced Compression nur als Option zur Enterprise Edition
- Ideal für relativ denormalisierte Daten (Typisch für Materialised Views)
- Update / Create Performance „mittelmäßig“ bei großen Änderungen
- Blockgröße in Überlegungen einfließen lassen
- Ideal für Log Tabellen

Material

- Metalink

- 9i R2 New Feature: Data Segment Compression
Doc ID: 228082.1
- 11g Table compression : Overview
Doc ID: 466362.1

- Links

http://www.oracle.com/technology/oramag/oracle/04-mar/o24tech_data.html

http://www.dba-oracle.com/oracle11g/sf_Oracle_11g_Data_Compression_Tips_for_the_DBA.html

<http://www.oracle.com/technology/products/database/oracle11g/pdf/advanced-compression->

http://www.oracle.com/technology/products/bi/pdf/o9ir2_compression_twp.pdf

http://weblog.infoworld.com/tcdaily/archives/2008/06/oracle_advanced.html



Oracle Table Compression



Gunther Pippèrr

gunther@pipperr.de
<http://www.pipperr.de>

Ihr Partner für Schulung, Betreuung und Beratung rund um die **Oracle-Datenbank**