



Tablespace Management

Johannes Ahrends
CarajanDB GmbH



DOAG
Deutsche ORACLE-Anwendergruppe e.V.

- **Experten mit über 20 Jahren Oracle Erfahrung**
- **Firmensitz in Erftstadt bei Köln**
- **Spezialisten für**
 - Oracle Datenbank Administration
 - Hochverfügbarkeit (RAC, Data Guard, Failsafe, etc)
 - Einsatz der Oracle Standard Edition
 - Oracle Migrationen (HW, Unicode, Konsolidierung, Standard Edition)
 - Replikation (Goldengate, SharePlex, Dbvisit)
 - Performance Tuning
- **Schulung und Workshops (Oracle, Toad)**



- **Oracle Spezialist seit 1992**
 - 1992: Presales bei Oracle in Düsseldorf
 - 1999: Projektleiter bei Herrmann & Lenz Services GmbH
 - 2005: Technischer Direktor ADM Presales bei Quest Software GmbH
 - 2011: Geschäftsführer CarajanDB GmbH
- **2011 → Ernennung zum Oracle ACE**
- **Autor der Bücher:**
 - Oracle9i für den DBA, Oracle10g für den DBA, Oracle 11g Release 2 für den DBA
- **Leiter der DOAG SIG Database**
- **Hobbies:**
 - Drachen steigen lassen (Kiting) draußen wie drinnen (Indoorkiting)
 - Motorradfahren (nur draußen)



- E-Mail: johannes.ahrends@carajandb.com
- Homepage: www.carajandb.com
- Adresse:
 - CarajanDB GmbH
Siemensstraße 25
50374 Erftstadt
- Telefon:
 - +49 (22 35) 1 70 91 84
 - +49 (1 70) 4 05 69 36
- Twitter: [streetkiter](#)
- Facebook: [johannes.ahrends](#)
- Blogs:
 - blog.carajandb.com
 - www.toadworld.com



Tablespace Verwaltung

A **tablespace** is a logical storage container for segments. Segments are database objects, such as tables and indexes, that consume storage space. At the physical level, a tablespace stores data in one or more data files or temp files.

A database must have the SYSTEM and SYSAUX tablespaces.

Quelle: Oracle Database Concepts 12c Release 1 Seite 12-33

- **Früher:**

- I/O Verteilung durch Trennung von Tabellen und Indizes
- Tabellengröße im MByte Bereich
- Plattengrößen im GByte Bereich
- Lokale Platten
- I/O Durchsatz ≤ 10 MB/s
- Striping durch Vergrößerung des Filesystems unausgewogen

- **Heute**

- S.A.M.E \rightarrow Stripe and Mirror Everything
- Tabellengrößen im TByte Bereich
- Festplattengrößen im TByte Bereich
- SAN / NAS Storage
- I/O Durchsatz ≥ 1 GB/s
- Rebalancing mit Automatic Storage Management

- **Früher:**

- Feste Größe einer Datei
- Begrenzte Anzahl von Datafiles (61)
- Maximal 121 Extents (2kB Blocksize)
- Extentverwaltung über FET\$ / UET\$
- Lücken in den Datafiles durch unterschiedliche Extentgrößen

- **Heute**

- Automatisches AUTOEXTEND bis „Unendlich“
- Anzahl Datafiles immer noch begrenzt (200)
- Locally Managed Tablespaces mit unlimitierter Anzahl Extents
- Locally Managed Tablespaces mit Extentverwaltung über Bitmap
- Locally Managed Tablespaces mit Autoallocate oder Uniform Size

- **Früher:**

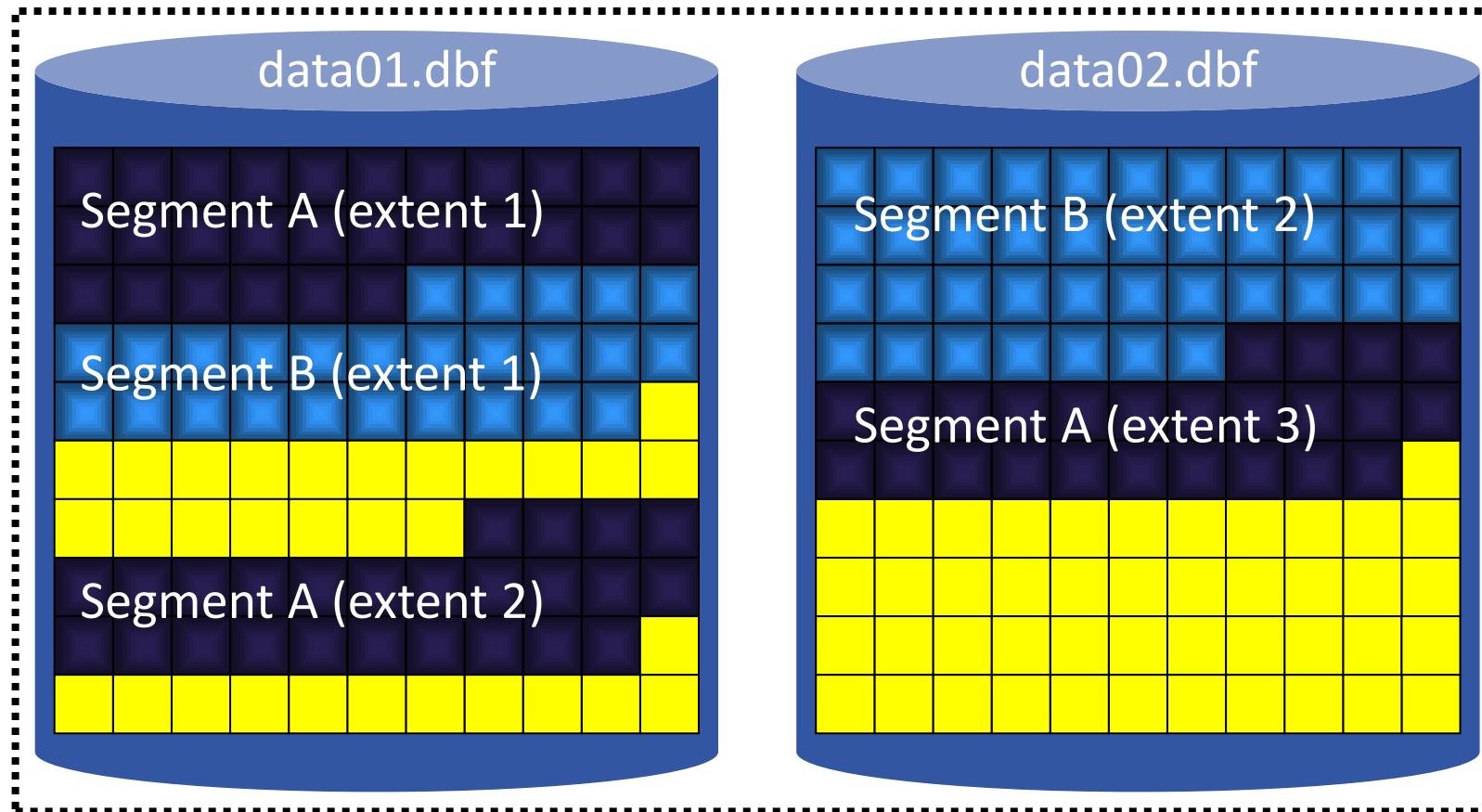
- Freelist-Konfiguration für OPS / RAC
- Fragmentierung im Block durch PCTUSED Parameter
- Fragmentierung nach Archivierung von Daten (DELETE FROM TABLE)
- PCTFREE zur Vermeidung von Migrated Rows




- **Heute**

- Automatic Segment Space Management
- Automatic Segment Space Management
- Automatic Segment Space Management
- PCTFREE zur Vermeidung von Migrated Rows

Tablespace (1)

Tablespace DATA



-  Datensegment
-  Indexsegment
-  Freier Block

- Verwaltungseinheit für alle Datenbankobjekte
- Bestehen aus mindestens einer Datendatei
- Permanente oder temporäre Datenspeicherung
- Erlauben es, Objekte zu gruppieren aus Administrations- oder Performancegesichtspunkten

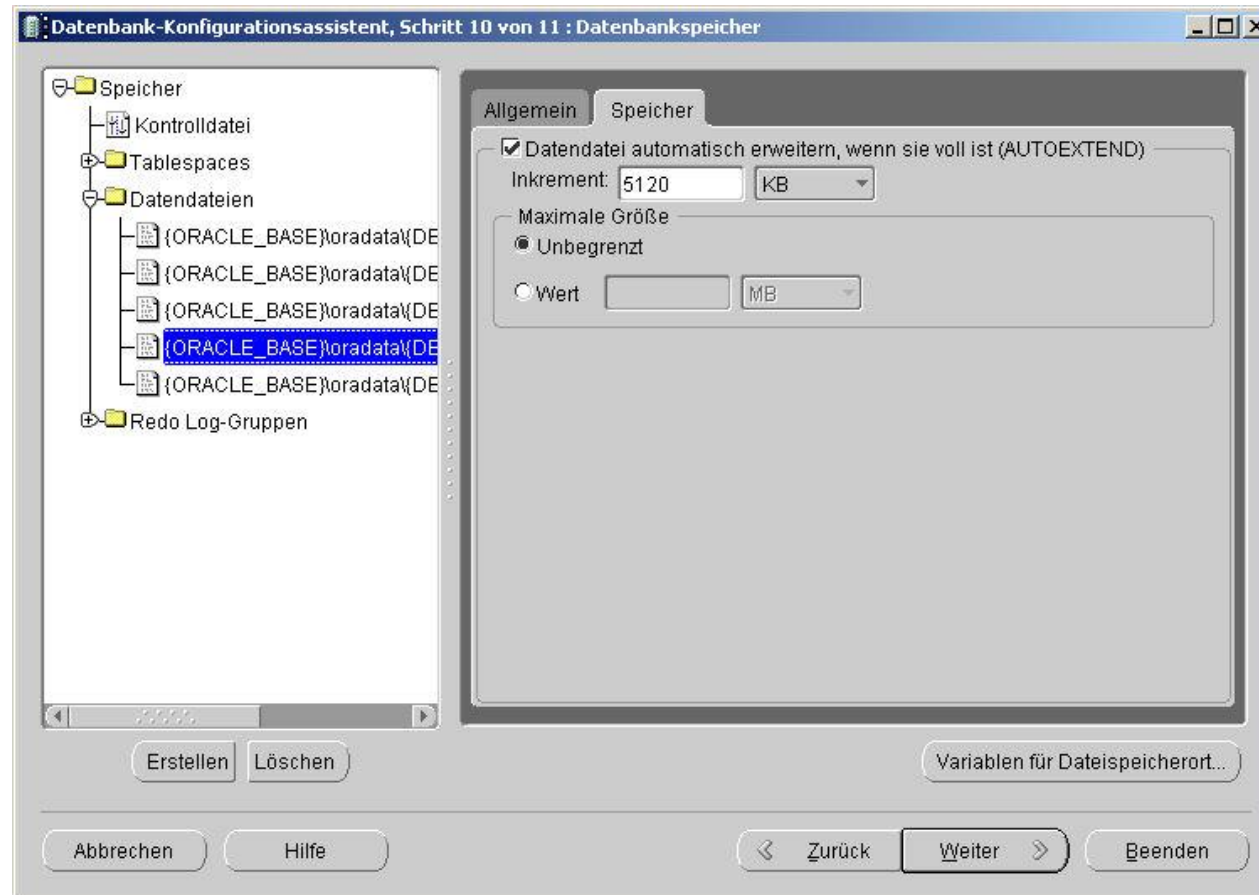
z.B:

- SYSTEM: Data-Dictionary
- UNDOTBS: Rollback Segmente
- TEMP: Temporäre Segmente (Sortierungen)
- DATEN: Große Tabellen
- INDEX: Indizes
- SYSAUX: Objekte für Tools (z.B: Enterprise Manager)

- **Tablespaces mit unterschiedlichen Blockgrößen möglich**
 - `CREATE TABLESPACE ... BLOCKSIZE 8K ...`
 - Für die Blocksize muss ein entsprechender Cache definiert sein:
Initialisierungsparameter `DB_n_CACHE_SIZE` ($n = 2,4,8,16,32$)
- **Spezialisierte Tablespaces**
 - TEMPORARY TABLESPACE
 - UNDO TABLESPACE
 - SYSAUX TABLESPACE (ab Oracle 10g)

Database Configuration Assistant

- In der Regel hat man nicht „unbegrenzt“ Platz!



- **Autoextend von Datafiles ist sinnvoll, wenn**
 - Eine Maximalgröße vorgegeben wird
 - Die Einheiten groß genug gewählt werden

- **Locally Managed Tablespace (Default)**
 - Extentverwaltung im Tablespace
 - Bitmap Struktur zur Verwaltung der Blöcke
 - Keine Rollback Information für DDL, da keine Änderung im Data Dictionary
 - Schnellere Extentverwaltung
- **Früher: Dictionary Managed Tablespace**
 - Extentverwaltung im Data Dictionary
 - Erzeugt wie jede DDL Operation Rollback Information
 - Management-Methode über `dbms_space_admin` änderbar
 - Vorsicht: Entspricht nicht zu 100% einem neu angelegten Tablespace!

- **AUTOALLOCATE = System Managed**
 - Initial Extent kann angegeben werden, Größen anderer Extents werden automatisch kalkuliert, mit einer Mindestgröße von 64 KByte pro Extent
- **UNIFORM**
 - Alle Extents haben die gleiche, konstante Größe
 - Storage-Klauseln können nicht angegeben werden

```
CREATE TABLESPACE <TS-NAME>  
...  
EXTENT MANAGEMENT LOCAL  
AUTOALLOCATE | UNIFORM SIZE n [K|M]
```

Automatic Segment Space Management

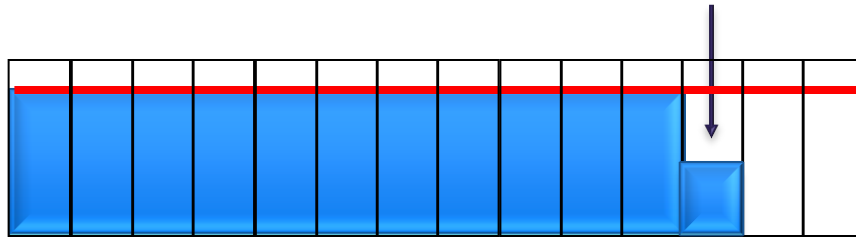
- **Belegter Platz pro Block wird in einer Bitmap abgelegt**
 - 5 Bit pro Block
- **Reduziert Freelist Contention**
- **Parameter PCTUSED wird damit überflüssig**
- **Migration nicht per Änderung bestehender Tablespaces möglich**
 - Datenmigration in neuen Tablespace notwendig (!)

- Freiplatz-Verwaltung („SEGMENT SPACE MANAGEMENT“)
- **MANUAL** = Freier Platz wird über sog. Freelists verwaltet
 - Tabellen im Data Dictionary
- **AUTO** = Freier Platz wird über Bitmap-Strukturen verwaltet (Automatic Segment Space Management, ASSM)
 - Schneller
 - Nur für Permanente Tablespaces
 - Nicht für SYSTEM- und Undo-Tablespace

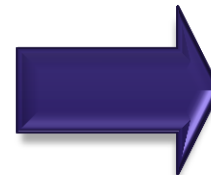
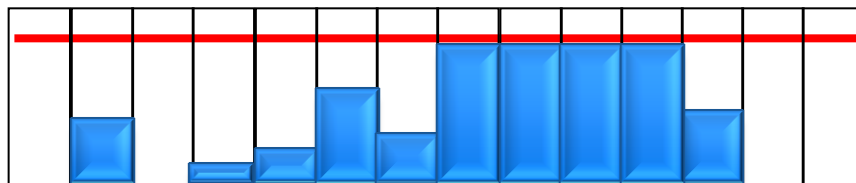
```
CREATE TABLESPACE <TS-NAME>  
...  
EXTENT MANAGEMENT LOCAL  
SEGMENT SPACE MANAGEMENT AUTO | MANUAL
```

Beispiel Archivierung + ASSM

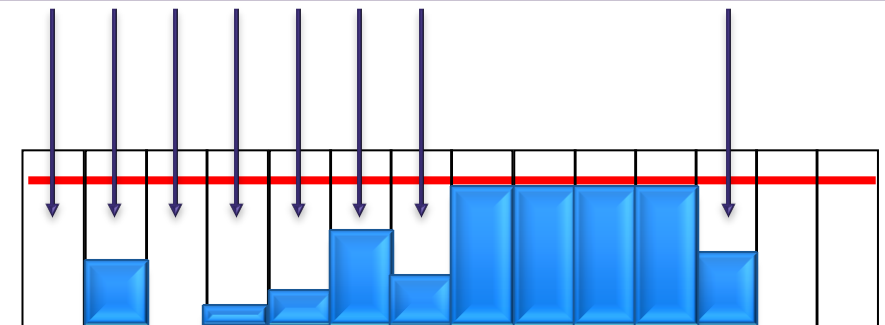
```
INSERT INTO AUFTRAEGE (...)  
VALUES ...
```



```
DELETE FROM AUFTRAEGE WHERE DATUM <  
sysdate - 90  
AND STATUS = 9
```

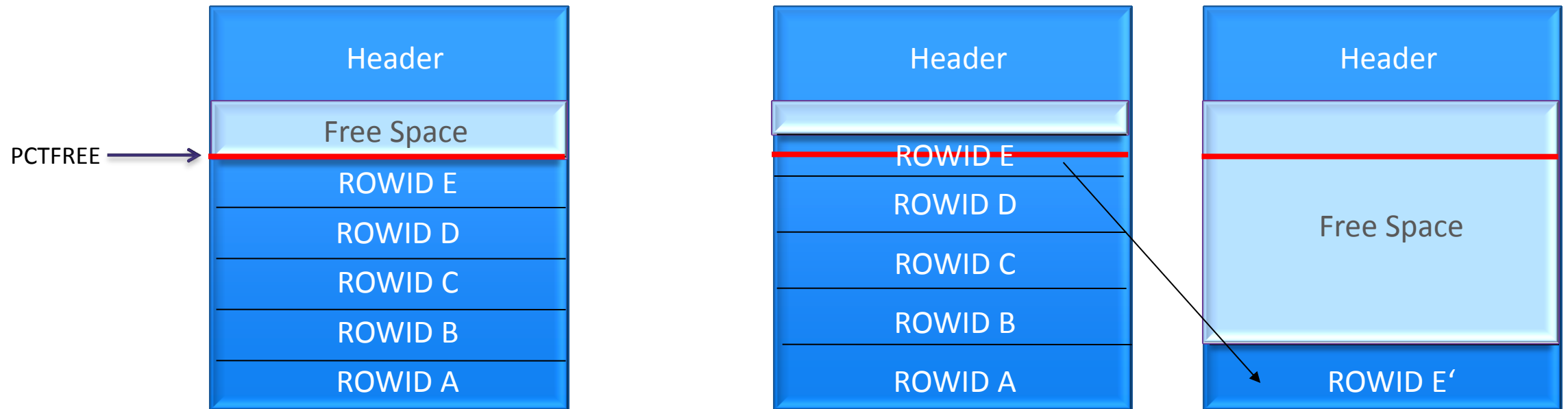


```
INSERT INTO AUFTRAEGE (...)  
VALUES ...
```



- **wichtiger Parameter für die Kapazitätsplanung**
 - Grenze für die Füllung von Datenblöcken mit INSERT
 - Default: 10% bleibt frei für UPDATE, ausreichend für die meisten Anwendungen
 - Kritisch, wenn:
 - Workflowprinzip => Einfügen von Daten über UPDATE
 - ALTER TABLE ADD COLUMN
 - Oracle 11g / 12c → Metadata-Only DEFAULT Wert für NULL Columns
- **Gefahr von Row Chaining als „Migrated Rows“**

Row Chaining – Migrated Rows





Reorganisation

- **Export/Import**
 - Immer noch im Einsatz
 - Gefährlich, da die Daten aus der Datenbank herausgeladen werden
 - Nur Offline möglich
 - Platz im Filesystem erforderlich (geringer als im Tablespace)
- **ALTER INDEX ... REBUILD (ONLINE)**
 - Effektivste Methode, einen Index zu reorganisieren
 - Online möglich (nur Enterprise Edition)
 - Zusätzlicher Platz erforderlich (im gleichen oder anderen Tablespace)
- **ALTER TABLE ... MOVE TABLESPACE**
 - Einfacher Befehl, um eine Tabelle zu reorganisieren
 - Nur Offline möglich (außer IOT)
 - Zusätzlicher Platz erforderlich (im gleichen oder anderen Tablespace)

- **ALTER TABLE ... SHRINK SPACE (CASCADE)**
 - Einfache Methode zur Reorganisation von Tabellen in Place
 - Nur bei Automatic Segment Space Management (ASSM)
 - Kein zusätzlicher Platz erforderlich
 - Einschränkungen (z.B. kein Securefile, Function based Index)
- **ALTER TABLESPACE ... SHRINK SPACE**
 - Nur temporary Tablespace (also keine wirkliche Alternative!)
- **DBMS_REDEFINITION**
 - Online Reorganisation von Tabellen über Prozeduren
 - Zusätzlicher Platz erforderlich
 - Einschränkungen (z.B: LONG -> LOB)

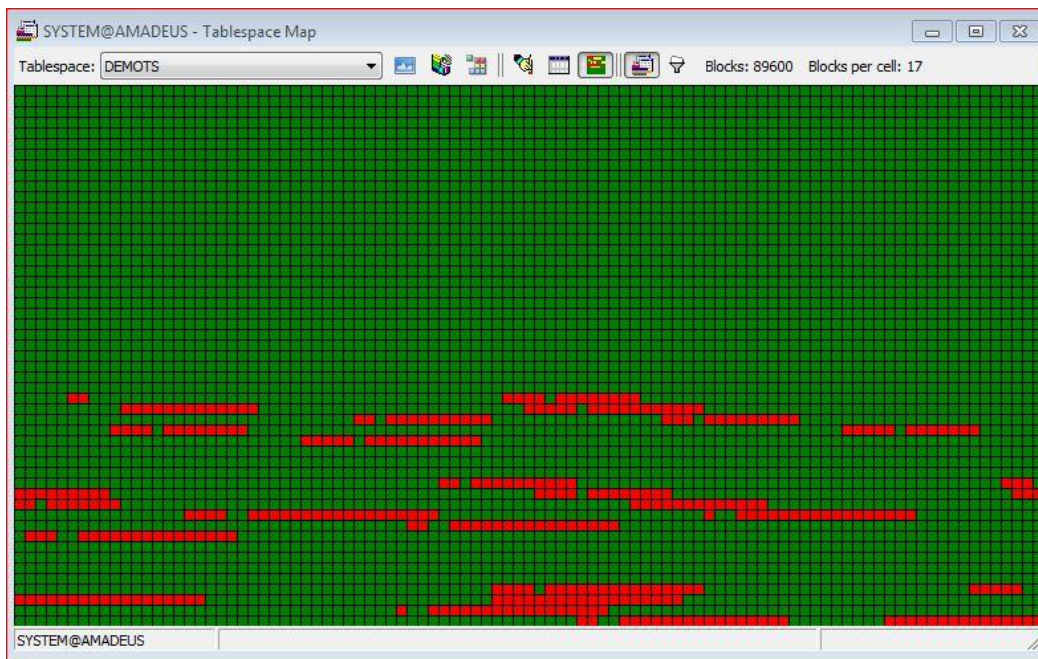
- **Offline (Auszeit proportional zur Segmentgröße)**
 - Menüpunkt Database → Optimize → Rebuild Multiple Objects
 - ALTER TABLE MOVE (online für IOTs bei Enterprise Edition)
 - ALTER INDEX REBUILD (online bei Enterprise Edition)
- **Offline Spezialfälle**
 - Menüpunkt Database → Optimize → Rebuild Table
 - Tabellen mit LONG-/LONG RAW-Spalten
 - Spaltenreihenfolge oder Spaltendatentypen ändern
 - Tabelle nach den Primärschlüsselspalten sortieren
 - Menüpunkt Database → Optimize → Repair Chained Rows
 - Beseitigung von Chained Rows

- Nur für große Tabellen sinnvoll
- Temporär ggf. erheblicher Platzbedarf
- **Effektive Methode:**
 - ALTER TABLE MOVE TABLESPACE
 - Gleicher Tablespace
 - Neuer Tablespace

MOVE im gleichen Tablespace

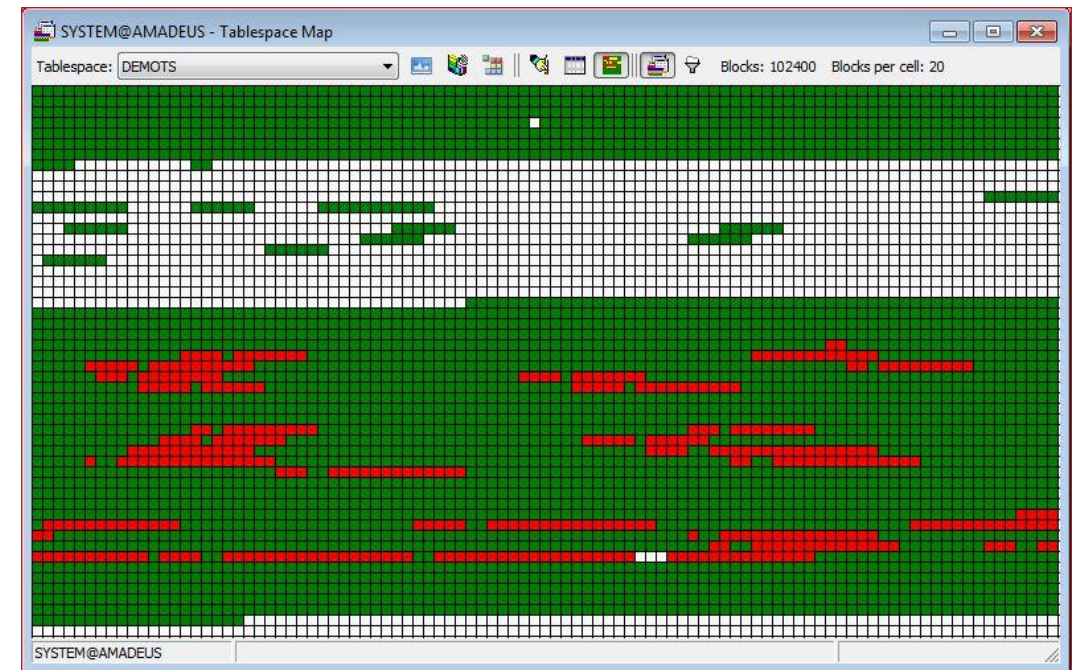
- **VORHER**

- 89600 Blocks
- 700 MB



- **NACHHER**

- 102400 Blocks
- 800 MB



Move Table Beispiel

```
SQL> CREATE TABLESPACE <tsneu> DATAFILE '<dfneu>' ...
SQL> ALTER TABLE <tabelle> MOVE TABLESPACE <tsneu>;
SQL> SELECT count(*) FROM dba_segments
      WHERE TABLESPACE_NAME = 'tsapp';

no rows selected

SQL> DROP TABLESPACE <tsapp> INCLUDING CONTENTS AND DATAFILES;
SQL> ALTER TABLESPACE <tsneu> RENAME TO <tsapp>;
SQL> ALTER TABLESPACE RENAME DATAFILE '<dfneu>' TO '<dfapp>';
```



Weitere Gründe

- **Spezielle Blocksize für einzelne Tablespaces**
 - Große Blocksize z.B. für Tabellen / Indizes mit VARCHAR2(32k)
 - Kleine Blocksize z.B. für LOB Objekte (1 LOB pro Block)
- **Archivierung**
 - Flashback Data Archive
 - Read Only Tablespaces
 - Verkleinern der Backup-Dateien
 - Verkürzen der Backup-Zeiten

- **Physikalische Kopie eines Tablespaces in eine andere Datenbank**
- **Limitierungen:**
 - Gleicher Zeichensatz (und NCHAR-Zeichensatz)
 - Keine Abhängigkeiten VON Objekten außerhalb des Tablespaces
 - Index mit Tabelle in anderem Tablespace
 - Andere Partitionen in anderen Tablespaces
 - Foreign Key-Constraints zu Tabelle in anderem Tablespace
 - Während der Migration Read-Only
 - Plattformwechsel über Konvertierung mittels RMAN
- **Metadaten-Übertragung (Data Dictionary Export/Import)**
 - `EXP TRANSPORT_TABLESPACE=y ...`
 - `IMP TRANSPORT_TABLESPACE=y ...`

- **RMAN Parallelisierung Enterprise Edition**
 - Bis Oracle 11g: Multisection Backup nur bei Full Backup
 - Ab Oracle 12c: Multisection Backup auch bei Incremental Backup und Backup Copy
- **Manuelle RMAN Parallelisierung bei Standard Edition**
 - Backup einzelner Tablespaces / Datafiles statt der gesamten Datenbank
- **Table Recovery**
 - Wiederherstellung einer Tabelle aus dem RMAN Backup
 - Auxiliary Database mit SYSTEM, SYSAUX, UNDO und Segment-Tablespace

- **Tablespaces sind immer noch notwendig!**
- **Separate Tablespace für große Tabellen / Indizes**
 - Vereinfacht die Reorganisation
- **Anwendungsspezifische Trennung von Segmenten**
 - Tablespace Point-In-Time-Recovery für dedizierte Anwendungen
 - Table Recovery schneller
 - Transportable Tablespaces für einzelne Anwendungen

- **21.09.2015 DOAG Community Day Datenbanken in Leipzig**
 - Thema „Datenbank Cloning“
 - Referenten (bis jetzt)
 - Actifio
 - Delphix
 - IBM



Fragen?

johannes.ahrends@carajandb.com

DOAG
Deutsche ORACLE-Anwendergruppe e.V.