Tablespace – völlig überflüssig!?

Johannes Ahrends – CarajanDB GmbH

Johannes Ahrends

- 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
- DOAG Themenverantwortlicher Datenbankadministration, Standard Edition
- DOAG Botschafter 2019
- Hobbies:
  - Drachen steigen lassen (Kiting) draußen wie drinnen (Indoorkiting)
  - Motorradfahren (nur draußen)
  - Singen (überall)

ORACLE ACE

CarajanDB

- Experten mit über 30 Jahren Datenbank Erfahrung
- Spezialisten für
  - Datenbank Administration (Oracle und PostgreSQL)
  - Hochverfügbarkeit (RAC, Data Guard, Replication, etc.)
  - Migrationen (Unicode, PostgreSQL, Multitenant)
  - Performance Optimierung
  - Monitoring (OEM, Foglight, CheckMK, PEM)
- Fernwartung
- Schulung und Workshops
  - PostgreSQL
  - Oracle Multitenant
  - Toad

Quest Silver Partner

ORACLE Partner

Dbvisit GOLD RESELLER

EDB POSTGRES ECOSYSTEM PARTNER GOLD

Tablespace Verwaltung

Definition

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.

In a CDB, each PDB and application root has its own set of tablespaces.

Every CDB root, PDB, and application root must have the SYSTEM and SYSAUX tablespaces.

Quelle: Oracle Database Concepts 21c August 2021

Permanent Tablespaces

A permanent tablespace groups persistent schema objects. The segments for objects in the tablespace are stored physically in data files. Each database user is assigned a default permanent tablespace.

A very small database may need only the default SYSTEM and SYSAUX tablespaces. However, Oracle recommends that you create at least one tablespace to store user and application data.

You can use tablespaces to achieve the following goals:

- Control disk space allocation for database data
- Assign a quota (space allowance or limit) to a database user
- Take individual tablespaces online or offline without affecting the availability of the whole database
- Perform backup and recovery of individual tablespaces
- Import or export application data by using the Oracle Data Pump utility
- Create a transportable tablespace that you can copy or move from one database to another, even across platforms

Quelle: Oracle Database Concepts 21c August 2021

CARAJAN DB

PostgreSQL

„Tablespaces in PostgreSQL allow database administrators to define locations in the file system where the files representing database objects can be stored. Once created, a tablespace can be referred to by name when creating database objects."

"By using tablespaces, an administrator can control the disk layout of a PostgreSQL installation. This is useful in at least two ways. First, if the partition or volume on which the cluster was initialized runs out of space and cannot be extended, a tablespace can be created on a different partition and used until the system can be reconfigured."

"Second, tablespaces allow an administrator to use knowledge of the usage pattern of database objects to optimize performance. For example, an index which is very heavily used can be placed on a very fast, highly available disk, such as an expensive solid state device. At the same time a table storing archived data which is rarely used or not performance critical could be stored on a less expensive, slower disk system."

Quelle: https://www.postgresql.org/docs/10/manage-ag-tablespaces.html

Früher und Heute

- 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 → 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 und Heute (2)

- Früher:
  - Feste Größe einer Datei
  - Begrenzte Anzahl von Datafiles (Default 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 (Default 200)
  - Locally Managed Tablespaces mit unlimitierter Anzahl Extents
  - Locally Managed Tablespaces mit Extentverwaltung über Bitmap (ASSM)
  - Locally Managed Tablespaces mit Autoallocate oder Uniform Size

Früher und Heute (3)

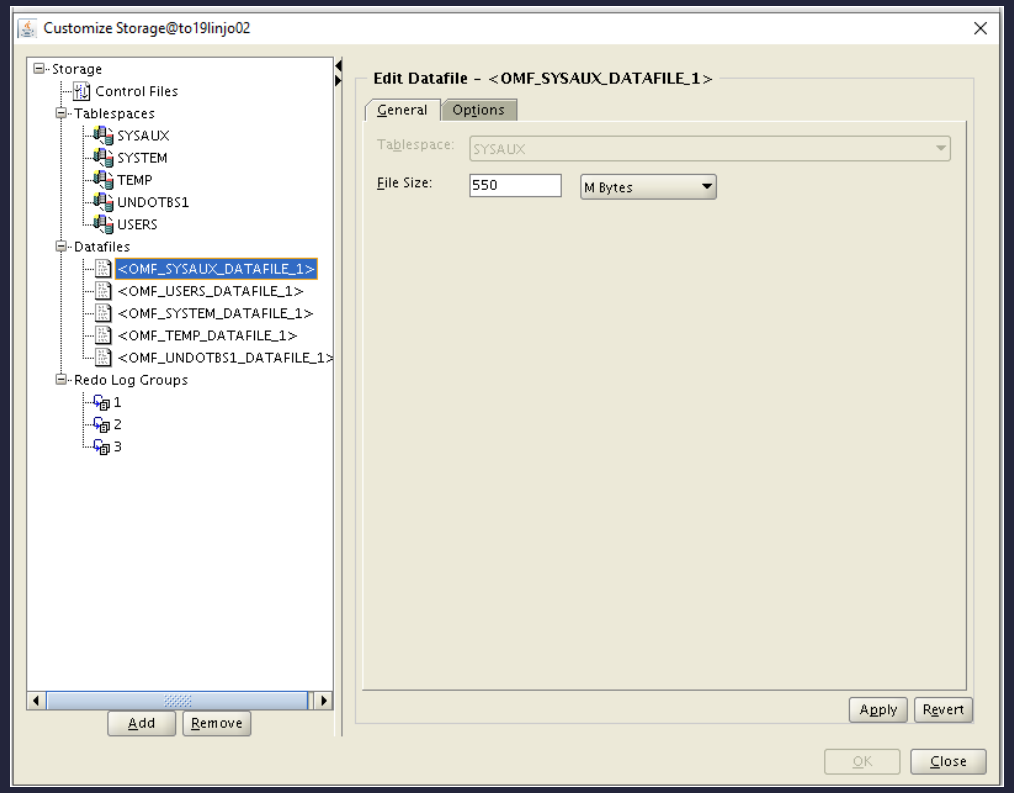
- Früher:
  - Freelist-Konfiguration für OPS / RAC
  - Fragmentierung im Block durch PCTUSED Parameter
  - Fragmentierung nach Archivierung von Daten (DELETE FROM TABLE)
  - PCTFREE zur Vermeidung von Migrated Rows
- Heute
  - Automatic Segment Space Management
  - Automatic Segment Space Management
  - Automatic Segment Space Management
  - PCTFREE zur Vermeidung von Migrated Rows

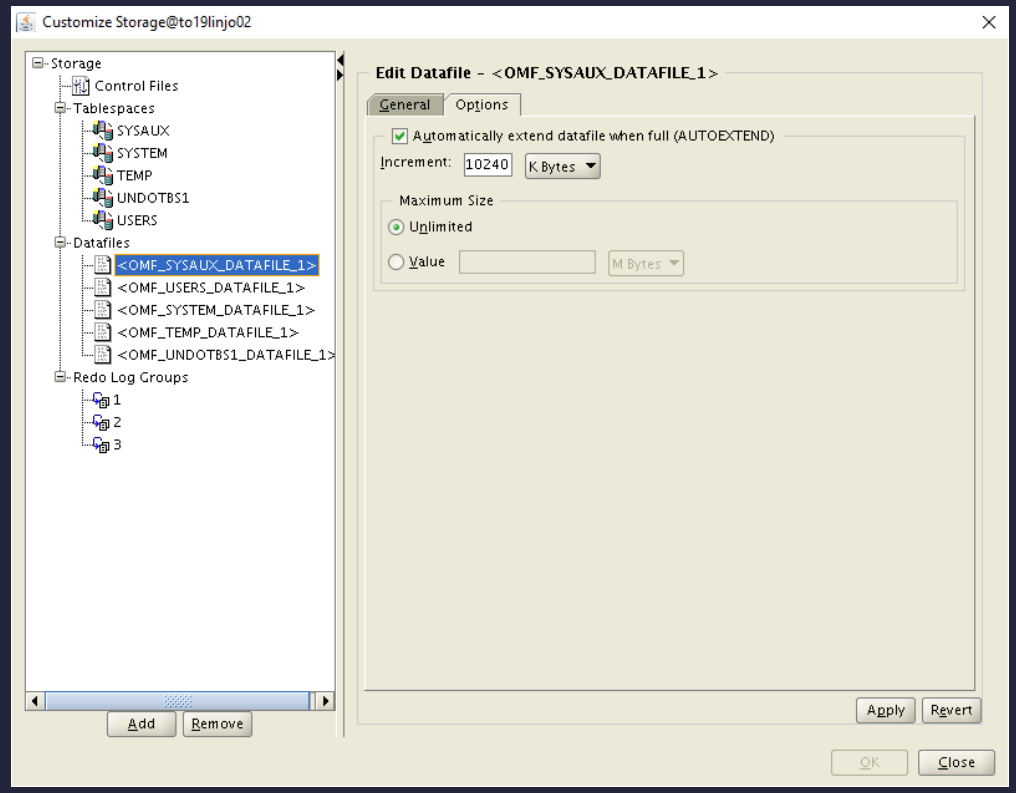
Tablespace-Optionen

- 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!





SYSAUX Tablespace

- Großes Wachstum im SYSAUX Tablespace!

```
SQL> SELECT tablespace_name, round(bytes/1024/1024/1024) GByte FROM dba_data_files;

TABLESPACE_NAME        GBYTE
------------------ ---------
SYSTEM                     1
SYSAUX                    25
UNDOTBS1                  88
UNDOTBS2                  22
USERS                      0
UNDOTBS3                 100
```

SYSAUX Tablespace (2)

- SYSTEM und SYSAUX Tablespace kann man nicht reorganisieren!

```
SQL> SELECT owner, segment_name, segment_type, round(bytes/1024/1024) MByte
       FROM dba_segments
      WHERE tablespace_name = 'SYSAUX'
      ORDER BY bytes DESC fetch first 10 rows only;

OWNER      SEGMENT_NAME                   SEGMENT_TYPE      MBYTE
---------- ------------------------------ ------------ ----------
SYS        WRH$_SYSMETRIC_HISTORY_INDEX   INDEX              3584
SYS        WRH$_SYSMETRIC_HISTORY         TABLE              2560
SYS        WRH$_RSRC_CONSUMER_GROUP       TABLE              1088
SYS        WRH$_BG_EVENT_SUMMARY_PK       INDEX               584
SYS        WRH$_RSRC_CONSUMER_GROUP_PK    INDEX               560
SYS        WRH$_SYSMETRIC_SUMMARY         TABLE               448
SYS        WRH$_BG_EVENT_SUMMARY          TABLE               432
SYS        WRH$_ENQUEUE_STAT_PK           INDEX               360
SYS        SYS_LOB0000006421C00005$$      LOBSEGMENT          328
SYS        WRH$_ENQUEUE_STAT              TABLE               264
```

CARAJAN DB

My Oracle Support

- Troubleshooting Issues with SYSAUX Space Usage (Doc ID 1399365.1)

Space Usage

Most issues with SYSAUX relate to the usage of space. The following articles can help in this regard together with the output from the awrinfo.sql script mentioned above to determine which components are using the most space:

Document 552880.1 General Guidelines for SYSAUX Space Issues
Document 1292724.1 Suggestions if your SYSAUX Tablespace grows rapidly or too large

Document 287679.1 Space Management In Sysaux Tablespace with AWR in Use

Document 1271178.1 Reduce SYSAUX Tablespace Occupancy Due to Fragmented TABLEs and INDEXes
Document 329984.1 Usage and Storage Management of SYSAUX tablespace occupants SM/AWR, SM/ADVISOR, SM/OPTSTAT and SM/OTHER
Document 1499542.1 Reducing the Space Usage of the SQL Management Base in the SYSAUX Tablespace

Document 556183.1 SYSAUX tablespace grows quite fast due to Apply spilling
Document 396502.1 High Storage Consumption for LOBs in SYSAUX Tablespace
Document 814710.1 WRI$_OPTSTAT_SYNOPSIS$ is rapidly filling the SYSAUX tablespace
Document 1243058.1 Wrh$_sql_plan table growth causes Sysaux Tablespace size increase continuously

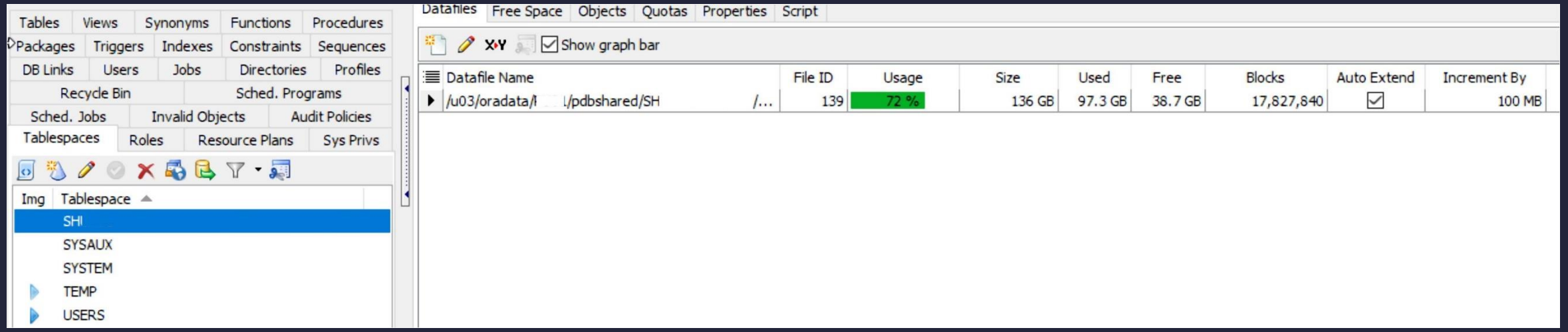
CARAJAN DB

Autoextend

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

Tablespace Full

Tablespace Full



- Aber da ist doch noch genug Platz …

Tablespace Full (2)

- Wie viele Stücke mit einer Größe von mind. 1 Mbyte

```
SQL> SELECT round(bytes/1024/1024) FROM dba_free_space
      WHERE tablespace_name = '<TS01>' AND bytes > 1024*1024;


ROUND(BYTES/1024/1024)
----------------------
                  1067
                  2048
                  2048
                    14
```

- Insgesamt nur ca. 5 GByte frei → wo sind die fehlenden ca. 32 GByte (38,7 – 5)?

Tablespace Full (3)

- Wie viele Stücke mit einer Größe von weniger als 1 MByte

```
SQL> SELECT count(*), round(sum(bytes)/1024/1024) MByte FROM dba_free_space
      WHERE tablespace_name = '<TS1>' AND bytes < 1024*1024;

  COUNT(*)       MBYTE
---------- ----------
     44132       34517
```

- Es gibt 44132 Lücken mit einer Gr´öße von weniger Als 1 MByte, macht insgesamt 34.517 MByte

Tablespace Full (4)

```
SQL> SELECT owner, segment_name, partition_name, segment_type, max(bytes)/1024/1024 MByte
       FROM dba_extents
      WHERE Tablespace_name = '<TS1>' AND bytes > 1024*1024
      GROUP BY owner, segment_name, partition_name, segment_type
      ORDER BY 5 desc,1,2,3
      FETCH FIRST 10 ROWS ONLY;

OWNER    SEGMENT_NAME           PARTITION_NAME SEGMENT_TYPE         MBYTE
-------- ---------------------- -------------- ------------------ ------
USERABC  ODS_AVDSFDDSFDSDFTEN                  TABLE                  64
USERABC  RAWSDFSDFSDFR_FK                      INDEX                  64
USERABC  SDFSDFSFFDDFSFDS                      INDEX                  64
USERABC  RASDFSDFDFSDSFIDX                     INDEX                  64
USERABC  RASDFDSFIBBUVWIDX2                    INDEX                  64
USERABC  PARTTABLEABC           SYS_P17212     TABLE PARTITION        64
USERABC  PARTTABLEABC           SYS_P18329     TABLE PARTITION        64
USERABC  PARTTABLEABC           SYS_P19704     TABLE PARTITION        64
USERABC  PARTTABLEABC           SYS_P21903     TABLE PARTITION        64
USERABC  PARTTABLEABC           SYS_P22932     TABLE PARTITION        64
```

CARAJAN DB

Tablespace-Management

- 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!

CARAJANDB

Locally Managed 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 (!)

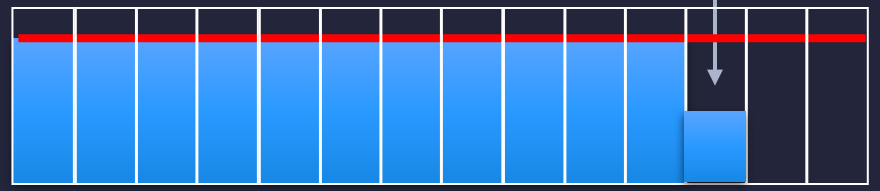
Automatic Segment Space Management

- 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

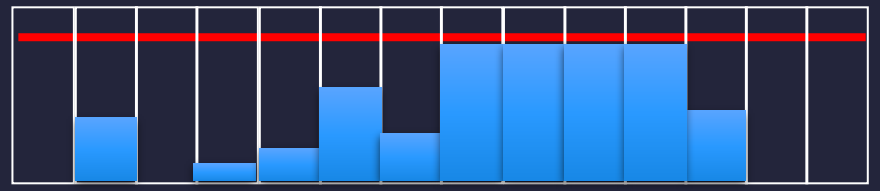
```
CREATE TABLESPACE <TS-NAME>
...
EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO | MANUAL
```

Beispiel Archivierung + ASSM

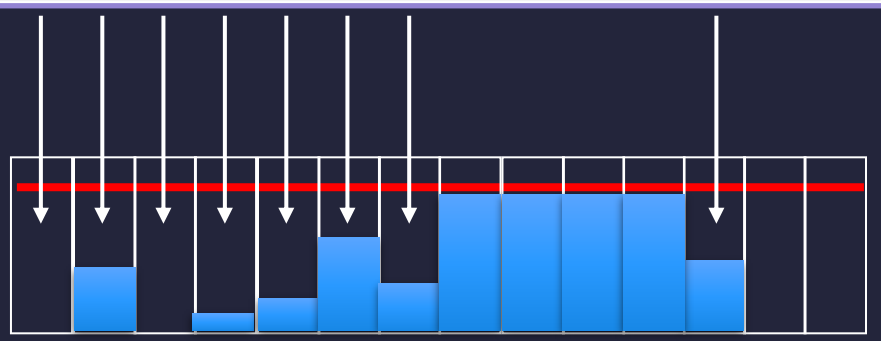
```
INSERT INTO AUFTRAEGE(...)
VALUES ...
```



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



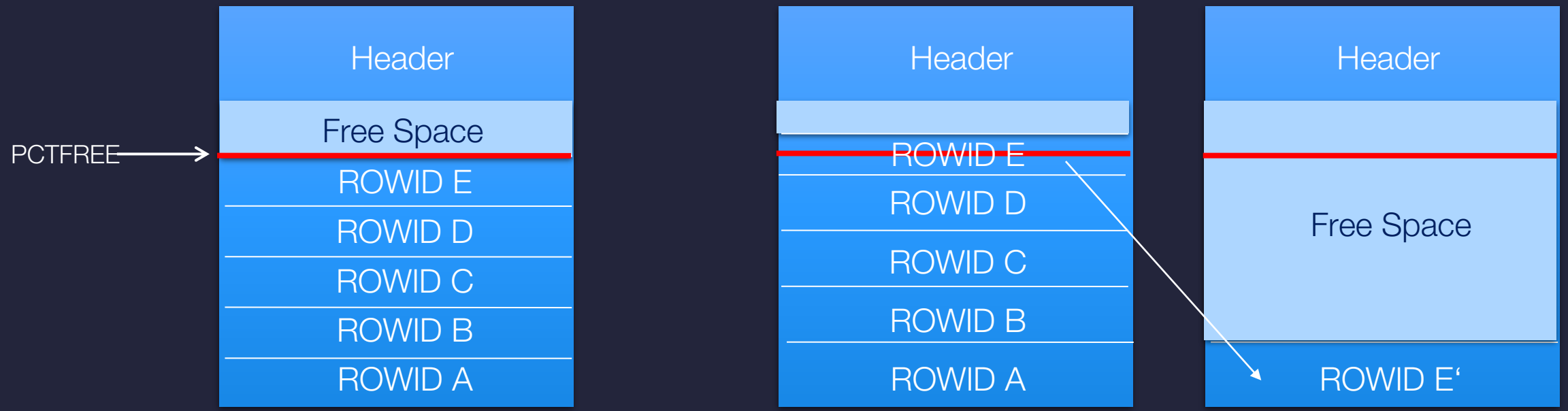
```
INSERT INTO AUFTRAEGE(...)
VALUES ...
```



PCTFREE

- 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

Reorganisationsmethoden (1)

- `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)

Reorganisationsmethoden (2)

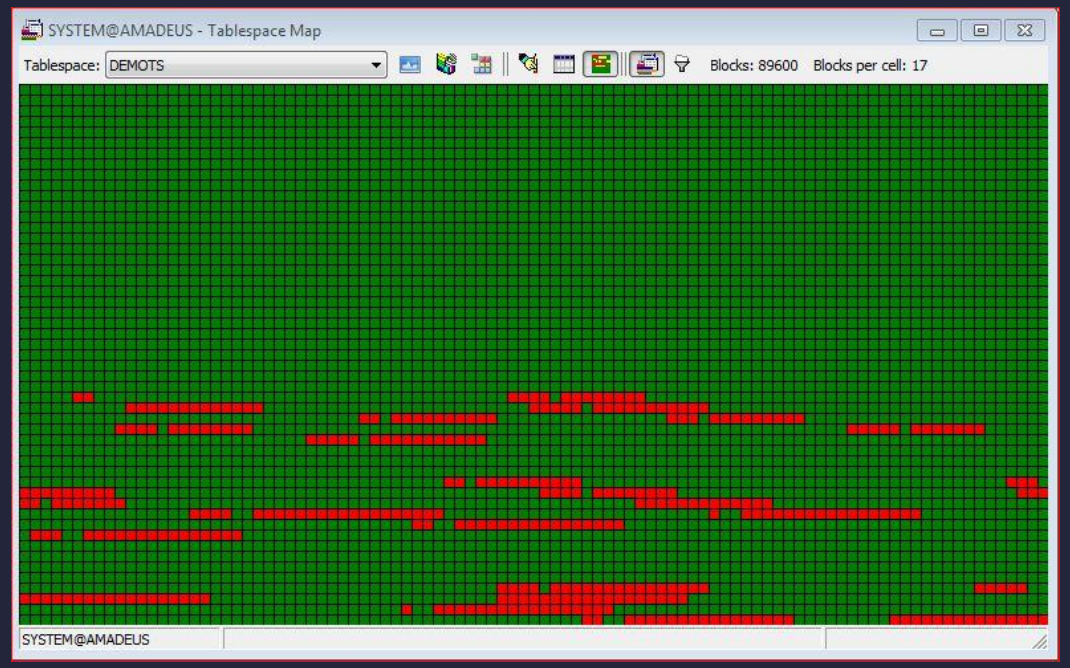
- `ALTER TABLE … SHRINK SPACE (CASCADE)`
  - Einfache Methode zur Reorganisation von Tabellen in Place
  - Nur bei Automatic Segment Space Management (ASSM)
  - Kein zusätzlicher Platz erforderlich
  - Einschränkungen (z.B. kein Securefile, Function based Index)
- `ALTER TABLESPACE … SHRINK SPACE`
 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

Reorganisation

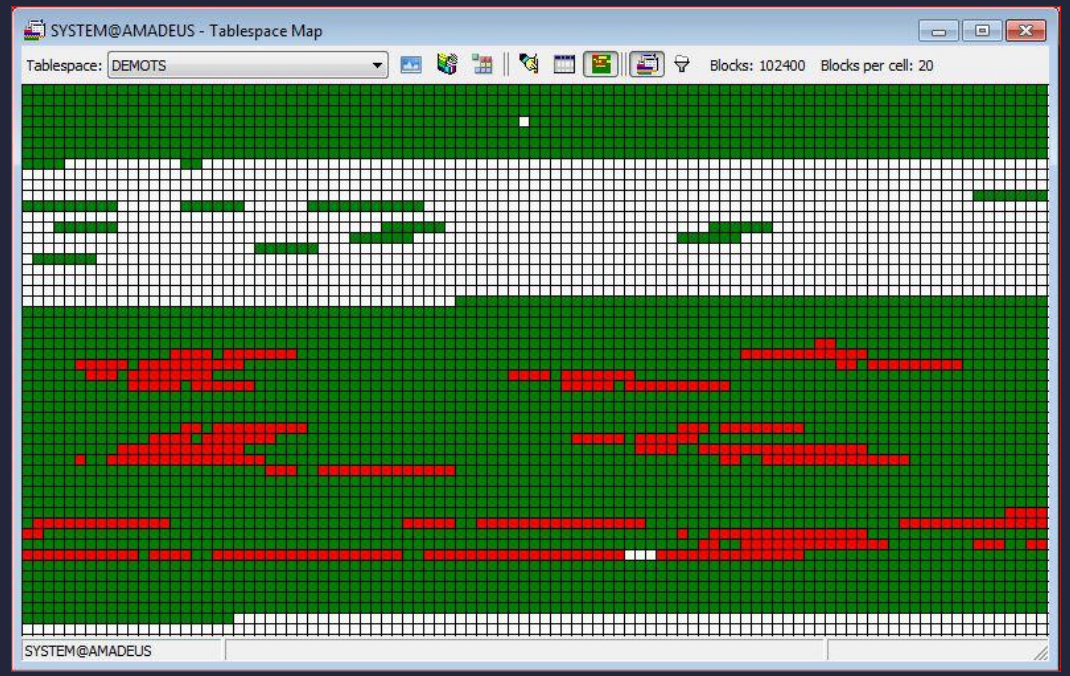
- Nur für große Tabellen sinnvoll
- Temporär ggf. erheblicher Platzbedarf
- Effektive Methode:
  - ALTER TABLE MOVE TABLESPACE
    - Gleicher Tablespace
    - Neuer Tablespace

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

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

Transportable Tablespace

- 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 ...`

CARAJAN DB

RMAN

- 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

Multitenant

- Ausgangslage:
  - NON-CDB mit Verwaltungsschemata mit eigenem Default Tablespace
    - Audit (audit)
    - Monitoring 1 (oemuser)
    - Monitoring 2 (paomon)
    - Backup (netbackup)
- Multitenant
  - jedes Schema wird zu einem Common Schema (C##)
  - Default Tablespaces müssen in allen PDBs angelegt werden!
  - Besser nur ein Default Tablespace für alle Verwaltungsschemata (Tools ;-))

CARAJAN DB

Fazit

- Tablespaces sind immer noch notwendig!
- Maximale Größe von Datafiles vorgeben (es gibt keinen unendlichen Space)
- 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

Kontakt

- E-Mail: johannes.ahrends@carajandb.com
- Homepage: www.carajandb.com
- Adresse:
  - CarajanDB GmbH
    Siemensstraße 25
    50374 Erftstadt
- Telefon:
  - +49 (1 70) 4 05 69 36
- Twitter: carajandb
- Facebook: johannes.ahrends
- Blogs:
  - blog.carajandb.com