



Tablespaces und Datendateien

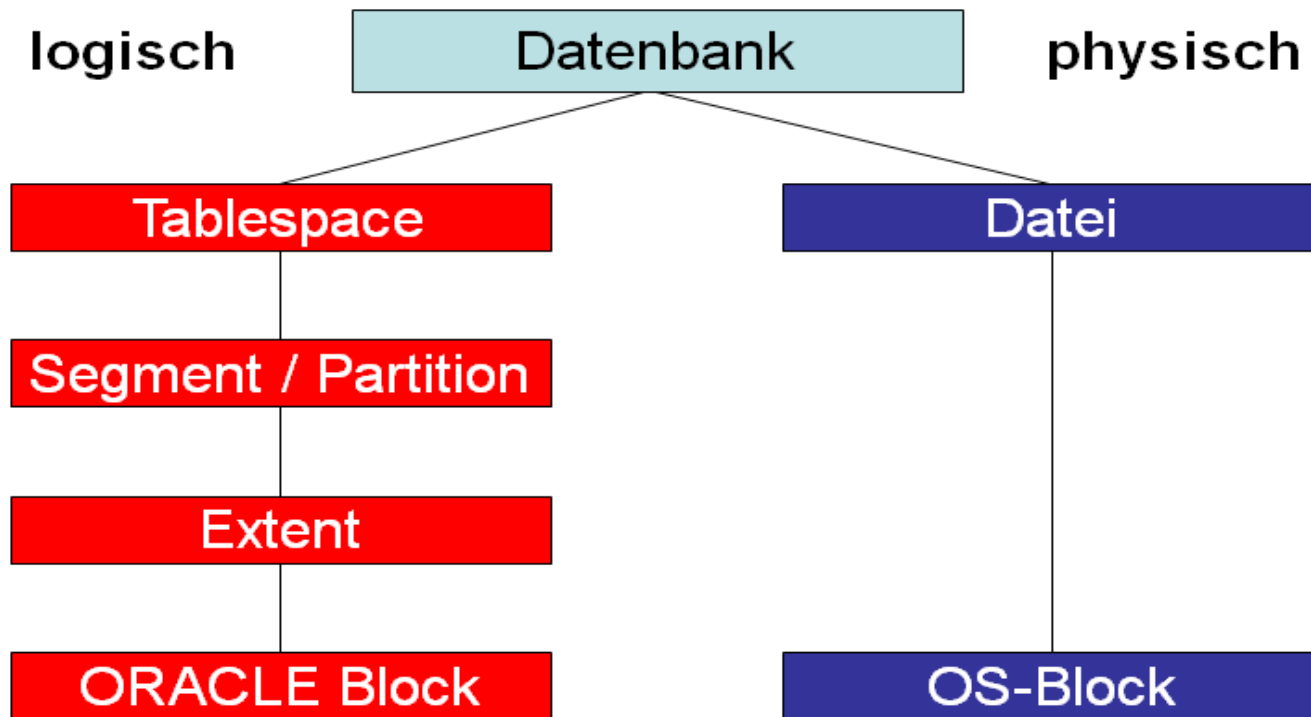
Thomas Klughardt
Senior Presales Consultant
thomas.klughardt@quest.com

Agenda

- Definition
 - Was sind Tablespaces und Datendateien?
- Planung
 - Warum sollte man frühzeitig darüber nachdenken?
- Eigenschaften
 - Parameter und Optionen
- Best Practices
 - Was ist wann sinnvoll?
- Fazit

Definition

- Ein Tablespace und dessen Datendateien sind die Brücke zwischen logischer und physischer Struktur.



Definition

- Sinn: Betriebssystemabhängige Strukturen verbergen
 - Instanzadministration ist OS abhängig
 - Schemaadministration ist OS unabhängig
 - Scripte, die unter Linux laufen, sollen auch unter Windows laufen.
- Tablespace
 - Logischer Container
 - Enthält Segmente, Extents, Oracle Blöcke
- Datendatei
 - Physikalischer Container
 - Enthält Betriebssystem-Blöcke
- Ein Tablespace enthält eine oder mehrere Datendateien.

Planung

- Gedanken beim Tablespacedesign
 - Logische Aufteilung der Anwendung
 - Aufteilung nach Zuständigkeiten
 - Pflegeintensive Objekte
 - Schnell wachsende Segmente
 - Schnell fragmentierende Segmente (Updates auf Tabellen/Partitionen, DML auf Indizes)
- Historische Punkte für Aufteilung bei Stripe and Mirror Everything (SAME) uninteressant
 - Trennung kleiner und großer Objekte
 - IO Lastbalancierung, nicht alle IO intensiven Objekte auf eine Platte

Planung

- Backup und Restore
 - Restore & Recovery auf Tablespaceebene
 - Objekte zusammenfassen, die konsistent zueinander sein müssen
 - Objekte in einen Tablespace, die nicht gesichert werden müssen
- Transportable Tablespaces für Ladejobs
- Read Only Tablespaces für statische Daten
- Frühzeitig Gedanken machen
 - Tablespacelayout ändern ist aufwändig (Zeichensatz auch)
 - Arbeit und Auszeit

Verplant – Zu spät?

- Offline Reorganisation
 - `ALTER TABLE <Tabelle> MOVE TABLESPACE <Tablespace>;`
 - Bei mehreren Objekten vielleicht besser exportieren/importieren
 - Data Pump (`dpexp` und `dpimp`)
 - Klassischer Export / Import (`exp` und `imp`)
- Online Reorganisation
 - DBMS_Redefinition
 - Vorsicht mit `LONG`, `BFILE`, User Defined Types!
 - Space Manager with Live Reorg
 - Third Party, Quest Software

Eigenschaften

- Die Eigenschaften von Tablespaces und Datendateien betrachtet man am besten zusammen.
- **TABLESPACE**
 - Art des Tablespaces
 - **PERMANENT** (default)
 - Persistente Segmente
 - **TEMPORARY**
 - Temporärsegmente für Sortierungen
 - **UNDO**
 - Undo Segmente (konsistentes Lesen, Rollback)

Eigenschaften

- **DATAFILE**
 - Name der Datei, oder Oracle Managed File (OMF)
- **SIZE**
 - Initialgröße, bei OMF Standard bei 100 MB
- **SMALLFILE / BIGFILE (ab 10g)**
 - **SMALLFILE** (default)
 - Mehrere Datendateien für Tablespace möglich
 - Maximalgröße 2^{22} Blöcke (bei 8k Blocksize 32GB = $2^{22} * 8$ KB)
 - **BIGFILE**
 - Nur eine Datendatei pro Tablespace
 - Maximalgröße 2^{32} Blöcke (bei 8k Blocksize 32 TB = $2^{32} * 8$ KB)
 - Bei BIGFILE Tablespaces Manipulation der Datendatei über TS Namen möglich

Eigenschaften

- **AUTOEXTEND**

- Gibt an ob Tablespace wachsen darf
- Wenn ja, sollte **NEXT** und **MAXSIZE** angegeben werden

- **BLOCKSIZE**

- Gibt an wie groß die Datenblöcke werden sollen
- Wird nichts angegeben, wird die Datenbankeinstellung genommen
 - (DBCA Datenbankdefault 8k)
- Hat Einfluss auf maximale Größe
- Beeinflusst Anzahl der IO Operationen, um z.B. an Zeile zu kommen

Eigenschaften

- **NOLOGGING / LOGGING / FORCE LOGGING**
 - **NOLOGGING / LOGGING** für Tablespace gibt nur den Default an
 - Wird beim Anlegen genutzt, Override möglich
- **NOLOGGING** schreibt auch Redo Log Information bei Änderungen, außer:
 - Bei **CREATE TABLE AS SELECT (CTAS)**
 - **INSERT /*+APPEND*/**
 - Performance Vorteile, aber kein Recovery möglich
- **FORCE LOGGING** erzwingt Redo Log Informationen
 - Auf Datenbank oder Tablespaceebene möglich

Eigenschaften

- **EXTENT MANAGEMENT LOCAL / DICTIONARY**
 - **LOCAL** (default)
 - Extentverwaltung findet im Tablespaceheader statt
 - Definitiv sinnvoll
 - **DICTIONARY** (legacy)
 - Extentverwaltung im Data Dictionary
 - Meist Altlasten
 - Wenn Datenbanken mit DBUA hochgezogen wurden
- **SEGMENT SPACE MANAGEMENT AUTO / MANUAL**
 - **AUTO** (ASSM, default seit 9i)
 - Verwaltung der Füllgrade der Blöcke über Bitmap im Extent
 - **MANUAL**
 - Verwaltung der Füllgrade über Freelists im Segment Header

Best Practices

- SYSTEM Tablespace gehört der Datenbank
 - Anwendung hat da nichts zu suchen
 - Vorsicht beim Arbeiten mit SYS oder SYSTEM
- Nicht zu viele Tablespaces anlegen, eindeutig benennen
 - Zwar verlockend, aber nicht Zielführend
- Namen Case insensitive
 - Case sensitive möglich, aber man muss nicht alles mitmachen...

Best Practices

- DB Blockgröße = OS Blockgröße (oder Vielfaches)
 - Nicht unnötig viele IO Operationen durchführen
 - Keine irrelevanten Daten lesen
- AUTOEXTEND immer mit NEXT
 - Sonst wird bei jedem Extend ein Block allokiert
 - Bei Oracle Managed Files 100M
 - Immer eine MAXSIZE setzen
 - Man hat einfach nicht unendlich Diskspace

Best Practices

- LOB Tablespaces
 - LOBs ohne ASSM
 - Unter 10g gab es Probleme bei LOBs und ASSM
 - Unter 11g behoben?
 - Blocksize klein wählen
 - LOB allokiert in Chunks
 - Chunks sind immer ein Vielfaches der Blocksize
 - LOBs sind oft gar nicht „Large“

Best Practices – Backup Special

- Offline Backup immer im Mountmodus und mit RMAN
 - Sonst weiß man nie, ob man alle Datafiles erwischt hat
 - Das merkt man dann beim Restore...
 - RMAN ist genau dafür da
 - RMAN ist Benutzerfreundlich
- Export / Import ist kein Backup
 - Restorezeit (Import) dauert zu lange
 - Konsistenter Export kann fehlschlagen
 - ORA-01555: Snapshot too old
 - Wird eventuell nicht bemerkt
 - Wenn doch, muss man trotzdem von vorne Anfangen
 - Inkonsistenter Export ist nicht Konsistent
 - Immer mit FLASHBACK_SCN

Fazit

- Es lohnt sich, sich mit dem Thema zu beschäftigen.
- Man kann sich viel Arbeit ersparen.
- Man kann Platz und Kosten sparen.
- Man kann IO Performance Bottlenecks und Contention vermeiden.

F & A



So finden Sie mich:

Quest Software Stand (der mit dem **Toad**)
zwischen Raum Tokyo und dem Restaurant

thomas.klughardt@quest.com



Vielen Dank!

thomas.klughardt@quest.com