3. Architektur eines DBS (Oracle)

aus Sicht des Datenbank Server Rechners

Connectivity Komponente(n) des DBS (z.B. Oracle Listener)

Installation
ORACLE_HOME

Instanz
ORACLE_SID

Datenbank

Oracle:
1 (aktive) Datenbank pro Instanz
Mehrere Instanzen pro DB möglich
(RAC bzw. Parallel Server)

3.1 Datenbank – Logische Sicht

3.1.1 Datenbankobjekte

organisiert in Schemata
(Namensraum, i.a. DB Objekte eines Benutzers)

Objekt Typen
Varrays
Nested Tables

Tabellen
Sequenzen
Views
Stored Procedures
Trigger
Database Links
Synonyme
…

3.1.2 Benutzerverwaltung

Oracle verwaltet User üblicherweise in der Datenbank
- **Alternative: externe Benutzerverwaltung wird genutzt**

Instanzprivilegien auf Betriebssystemebene (Gruppen osdba, osoper) bzw. auf DBS Ebene via Password File (anzulegen mit der Utility orapwd, REMOTE_LOGIN_PASSWORDFILE Parameter ist auf exclusive zu setzen):
- **sysdba**
  - **=> create database**
- **sysoper**

Datenbank Systemprivilegien (Rollen)
- **connect**
- **resource**
- **dba**
- **...**

Datenbank Objektprivilegien
- **werden (ebenfalls) mit grant vergeben und mit revoke zurückgenommen:**
  - **grant Privilegien on DB Objekt to Privilegnehmer**

- **Objektprivilegien abhängig von DB Objekten**
- **z.B.**
- **Tabelle=> select, insert, update, delete, alter, index, references, all**
- **View => select, insert, update, delete, all**
- **Stored Procedure=> execute**
- **Objekt Typ=> execute**

Rollen
- **Privilegnehmer und Privileg**
  - **grant Privileg to Rolle**
  - **grant Rolle to Privilegnehmer**
- **Benutzer haben Default Rollen (beim create / alter user festgelegt)**
- **Rollenwechsel möglich (set role)**

Privilegnehmer: Benutzer, Rollen, public

Profile (Quoten)

3.1.3 Systemkatalog

user_ eigene Datenbankobjekte
all_ alle Datenbankobjekte, auf die Zugriff besteht
dba_ alle Datenbankobjekte (privilegierter Zugriff)

v$ Instanzinformation (privilegierter Zugriff)

3.2 Instanz

3.2.1 Administration

Startup / Shutdown

sqlplus / nolog

connect internal oder connect ... as sysdba / sysoper
startup nomount / mount /open
shutdown normal / transactional / immediate / abort

lsnrctl => Listener Administration (start, stop, ...)

3.2.2 Prozesse

SQL verarbeitenden Prozesse:
- **Dedicated vs. Shared / Multithreaded Server**
- **Dispatcher**
- **Shared Server Prozesse**

Hintergrundprozesse:
- **Process Monitor (PMON)**
- **System Monitor (SMON)**
- **DB Writer (DBWn)**
- **Log Writer (LGWR)**
- **Checkpoint (CKPT)**
- **Archiver (ARCn)**
- **...**

3.2.3 Memory

PGA (Program Global Area)

SGA (System Global Area)
- **Buffer Cache(s)**
- **Redo Log Buffer**
- **Shared Pool**
  - **Row Cache (Data Dictionary)**
  - **Library Cache (QEPs)**
- **Java Pool**

Monitoring / Tuning:
- **Hit Rate beobachten**
- **Memory Komponente vergrößern, Paging beachten!**

3.3 Dateien

- **Datendateien (Datenbank)**
- **Control Files**
- **Online Redo Logs**
- **Archived Redo Logs**
- **Konfigurationsdateien**
  - **spfile**
  - **pfile**

3.4 Datenbank - Physische Sicht

Tablespace

Container für Datenbank Objekte

Typen von Tablespaces
- **System**
- **Permanent**
- **Nologging**
- **Temporär**
- **Undo / Rollback**

Default Tablespaces auf System- und Benutzerebene

Default Parameter für Speicherstrukturen auf Tablespace Ebene überschreiben System Defaults

Local vs. Dictionary Management

Data Files

Storage Units, aus denen Tablespaces bestehen

Block

Blockgröße für Datenbank festgelegt, für einzelne Tablespaces überschreibbar
- **=> Bezug zum Buffer Cache**

Segment

Teil eines Datenbank Objekts, der in einem Tablespace liegt

Typen von Segmenten:

- **Tabelle**
- **Index**
- **Partition**
- **Cluster**
- **Lob**
- **Lob Index**
- **Temporär**
- **Undo / Rollback**

Parameter für Speicherstrukturen überschreiben System bzw. Tablespace Defaults

Extent

Zusammenhängende Teilmenge eines Segments

(Def. von INITIAL, NEXT, MINEXTENTS, MAXEXTENTS, PCTINCREASE auf Tablespace- und Segmentebene)

Row

ROWID => Logische Adresse einer Zeile
(fix bis zu einer Reorganisation)

Migration / Chaining:

Falls eine Zeile (nach einem Update) nicht mehr in einen Block passt (aufgrund der variablen Zeilenlänge möglich), muß sie in einen anderen Block verschoben oder auf mehrere Blöcke aufgeteilt werden

Fixe ROWID => Indirekter Zugriff auf die Zeile

Vermeidung durch geeignete Def. von PCTFREE / PCTUSED (auf Segmentebene)

4. Physisches Datendesign und Query Optimierung

4.1 Physisches Datendesign

RAID

Partitionierung

4.2 Indexdesign

Btree Index
reverse
compress

Index Organized Tables

Cluster (Btree / Hash)

Function Based Index

Bitmap Index
Bitmap Join Index

4.3 Query Optimierung

4.3.1 Allgemein

Environment (OLTP / OLAP)

kleine / große Datenmengen, auf die zugegriffen wird
viele / wenige parallele Zugriffe
Read Write / Read Load
Zielsetzung (Antwortzeit / Durchsatz)

⇨ **Einfluß auf Strategien des Optimizers**

Einflußmöglichkeiten

Physisches Datendesign
Statistiken
optimizer_mode
Hints

Hilfsmittel

Analyse von QEPs (Query Execution Plan)

4.3.2 Operationen bei der Query Optimierung

Grundlage: Relationale Algebra

Restriktion
Projektion (Spaltenextrakt + Entfernung von Duplikaten)

Vereinigung (Hintereinanderschalten + Entfernung von Duplikaten)
Durchschnitt => Binäre Zuordnung (Equi Join)
Mengentheoretische Differenz => Binäre Zuordnung (Equi Join)

Kartesisches Produkt
Join => Equi Join, Non Equi Join
Quotient => kein SQL Konstrukt

Zusätzlich aus SQL:

Group by => Gruppierung (beinhaltet Duplikat Entfernung)
Having => Restriktion
Order by => Sortierter Zugriff

Subqueries, Views => Query Rewrite, falls möglich

4.3.3 Lokale Entscheidungen des Optimizers

Restriktion

- **Sequentiell**
- **Indexzugriff (speziell Index Only, Index Organized Table, Cluster)**

Equi Join

- **Nested Loop**
- **Sort Merge**
- **Hash**
  - **Wie sieht es mit Outer Joins aus?**
- **Index Join**
- **“Join Index“**
- **...**

Gruppierung

- **i.a. Sortierung / sortierter Zugriff**
- **Alternative: Hash Algorithmen**

Sortierter Zugriff

- **Sortieren oder sortierter Zugriff via Index**

4.3.4 Globale Entscheidungen des Optimizers

Berücksichtigung der Reihenfolge der Operationen (exponentielle Komplexität!)

Heuristiken für Antwortzeitoptimierung:

Pipelining

Vermeidung (aufwändiger) blockierender Aktionen:

Sortierung
Aufbau von Hash Tabellen
Materialisierung von Zwischenergebnissen

⇨ **Möglichst Indexzugriff für Restriktionen, (Nested Loop) Joins, sortierten Zugriff**

Allgemeine Heuristiken:

1. Restriktionen / Projektionen auf möglichst kleine Ergebnismengen möglichst früh, Zusammenfassung der Operationen

Zu beachten: Restriktionen haben zur Folge, dass die Indizes, über die die Restriktion unterstützt wird, für das Resultat nicht mehr zur Verfügung stehen

2. Wahl der geeigneten Zuordnungs Reihenfolge und Strategie

Zu beachten: Es bestehen Wechselwirkungen zwischen Reihenfolge der Operationen und der Zuordnungs Strategie!

3. Gruppierung / Entfernung von Duplikaten / Sortierung möglichst spät

4. Wiederverwendung von Zwischenergebnissen
(auf Transaktions- bzw. Sessionebene temporäre Tabellen einsetzbar)