

**mitp**

Dietmar Aust • Denes Kubicek • Jens-Christian Pokolm

# Oracle APEX und Oracle XE in der Praxis

Zahlreiche Best Practices  
zum sofortigen Einsatz

Formulare, Berichte, Benutzerverwaltung

Administration, Entwicklung, Deployment,  
Tuning, Debugging, Sicherheit

# Grundsätzliches zur Datenbank & Architektur von Oracle XE

Um auf den nachfolgenden Seiten das gleiche Verständnis der Begriffe zu gewährleisten, soll zunächst einmal den Aufbau einer Oracle-Datenbank dargestellt werden. Sie können dieses Kapitel auch überspringen, wenn es Ihnen noch zu sehr in »Elementarteilchen« der Datenbank geht – und erst später hierher zurückkehren.

## 1.1 Bestandteile der Datenbank

Das Datenbanksystem besteht aus einer Vielzahl von Datenbankdateien, auf die wir später noch im Detail kommen. Diese Sammlung von Dateien nennt man »Database«. Zur Verwaltung und Nutzung dieser Database läuft im Speicher des Datenbankservers entsprechende Verwaltungssoftware (dies kann natürlich auch Ihr Windows/Linux-Desktop-PC sein). Dieses Programm mit dem zugehörigen Hauptspeicher wird »Instanz« genannt. Eine Instanz setzt sich dort aus der so genannten SGA (System Global Area) und der PGA (Program Global Area) zusammen.

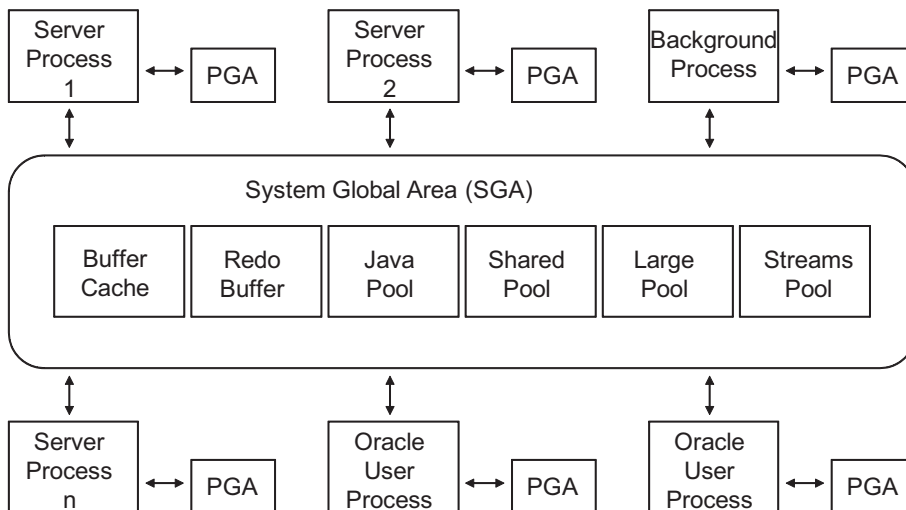


Abb. 1.1: Speicherbereiche der Oracle-Datenbank-Software im Überblick

In der SGA finden wir verschiedene Zwischenspeicher und Kontrollstrukturen für:

- Shared Pool (öffentlicher Anteil genutzter SQL-Statements zur möglichen Wiederverwendung)
- Data Dictionary Cache (interne Verwaltungsstruktur)
- Library Cache (Bibliotheksspeicher für PL/SQL und »parsed SQL«)
- Database Block Buffer (Cache für häufiger genutzte Datenblöcke)
- Redo-Log-Buffer (Speicher für Änderungslogbuch der DB) – Aus diesem Speicherbereich werden die so genannten ArchiveLogs erzeugt.
- Java (Code und Daten der JVM in der Datenbank)
- Large Pool (Session-Speicher für Shared-Server/MTS und interne Backup/Recovery-Operationen) – nicht unbedingt erforderlich

Der Speicher der SGA wird beim Start der Datenbank vom Betriebssystem angefordert und erst nach Beendigung wieder zurückgegeben. Schlägt diese Anforderung fehl, so startet die Datenbank nicht – z.B. mit der Fehlermeldung `unable to allocate xxx memory` oder `unable to attach to shared memory segment`. In diesem Fall kann man wahlweise den Hauptspeicher des Systems vergrößern oder die Konfigurationsparameter der Datenbank (`sga_target` bzw. `sga_max_size`) nach unten anpassen.

Die PGA enthält:

- Private SQL-Area, sessionbezogen (privater Anteil genutzter SQL-Statements, Bindevariablen und Laufzeitspeicher)
- Session Memory (Informationen über den User sowie die Art der Session – `dedicated/shared`)
- SQL-Work-Area (Sortierspeicher, `create & merges` von Bitmaps sowie Hash-Joins)

Wenn zumindest die SQL-Work-Area nicht vollständig in den Hauptspeicher des DB-Servers passt, so werden diese teilweise auf die Festplatte (Temp-Tablespace) ausgelagert. Dies ist jedoch um große Faktoren langsamer und es gilt, genau dieses möglichst zu vermeiden. Erreichbar ist das über einen möglichst gut ausgebauten Hauptspeicher sowie geschickte SQL-Statements.

Zugriffe und Operationen über gut gewählte Indizes vermeiden oftmals komplexe Sortiervorgänge und erhöhen die Performance spürbar.

Um eine Datenbank mit Tausenden aktiven Usern betreiben zu können, besteht die Möglichkeit, auf weiteren Systemen zusätzliche Instanzen zu betreiben (hori-

zontale Skalierung). Dabei werden die Datenbankdateien auf so genanntem »shared-Storage«, also einem von allen Systemen gleichberechtigt zugreifbaren Festplattenspeicher abgelegt. Dies kann bei handelsüblichen PCs entweder ein NFS-/NAS-Speicher oder auch eine per FireWire von mehreren Rechnern aus zugreifbare Festplatte sein.

### Hinweis

Für Umsteiger von Microsoft-SQL-Server oder MySQL: Üblicherweise wird bei MS-SQL & MySQL für eine neue Applikation eine neue Datenbank angelegt. Unter Oracle ist dies nicht erforderlich, sondern es werden in der Regel neue User angelegt und somit über das implizit vorhandene neue saubere Schema getrennt.

## 1.2 Datenbankprozesse

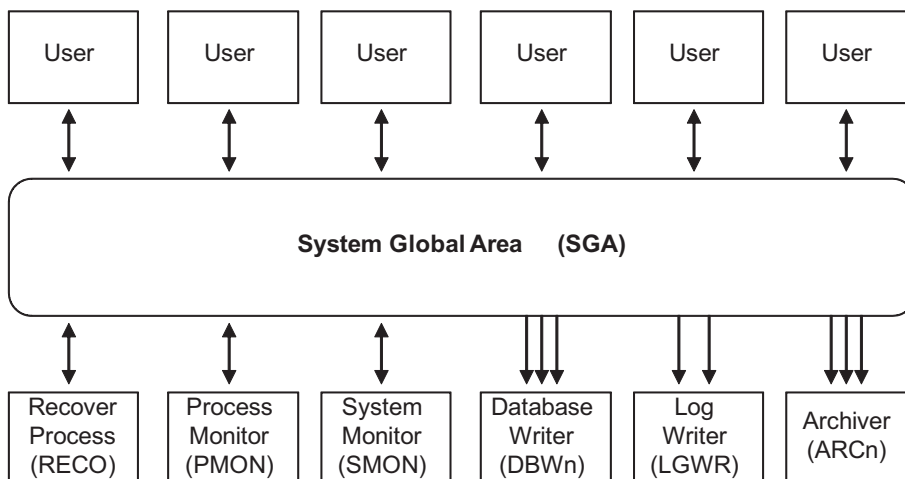


Abb. 1.2: Prozesse der Oracle-Datenbank-Software im Überblick

Unter MS-Windows kann man die unter Abbildung 1.2 aufgezeigten Prozesse leider nicht so leicht beobachten – aber diese sind dennoch vorhanden. Bei Unix/Linux/Mac OS hingegen werden die Prozessstrukturen durch das System-Kommando `ps -ax` sofort und einfach sichtbar.

Die oben in der Grafik angeordneten Benutzerprozesse (in der Regel für jeden angemeldeten User ein Prozess) kommunizieren mit der SGA des Systems. Die Hintergrundprozesse (unten) kümmern sich um den Verwaltungsaufwand und die Konsistenz der Datenbank.

Der Recover-Prozess ist dafür verantwortlich, bei verteilten Transaktionen diese zu überwachen und ggf. zu wiederholen.

Durch den LogWriter werden die Änderungen im Redo-Log-Buffer auf die Festplatte geschrieben. Wenn das somit geschriebene Online-Redo-Logfile dann voll ist, greift der Archiver-Prozess ein und archiviert das Online-Redo-Log an eine separate Stelle der Festplatte – diese Dateien nennt man dann »Archivelogs«.

Mit ihnen kann man im Fehlerfall eine Datenbank durch ein altes Backup der DB und dem »Einspielen« aller seit dem Sicherungszeitpunkt angefallenen Archivelogs vollständig auf einen frei wählbaren Zeitpunkt (zwischen dem Zeitpunkt des Backups und dem Fehlermoment) wieder herstellen. Dafür ist es erforderlich, dass alle benötigten Archivelogs konsistent verfügbar sind. Daher wird oftmals empfohlen, die Archivelogs auf einer anderen Festplatte als die eigentlichen Datendateien zu speichern. Dadurch kann selbst bei Verlust der Datenfestplatte die Datenbank bis kurz vor den Ausfallzeitpunkt lückenlos wiederhergestellt werden.

Der Archiver-Prozess ist jedoch nur bei Datenbanken vorhanden, die im »Archivelog-Mode« betrieben werden. Dies ist jedoch fast immer dringend zu empfehlen.

Der Database-Writer-Prozess schreibt die geänderten Blöcke aus dem Database Buffer Cache in die Datendateien zurück und prüft zugleich die Checksummen der geschriebenen Datenblöcke auf Konsistenz. Bei Systemen mit einer hohen Änderungslast kann man mehrere (n) dieser Prozesse starten.

Der SystemMonitor(SMON)-Prozess der Datenbank verwaltet die Speicherstrukturen der SGA und schiebt freie benachbarte Speicherblöcke zusammen. Darüber hinaus gibt er nicht mehr benötigten Platz im Temp-Tablespace (für temporäre Operationen wie z.B. das Sortieren) frei. Alle anderen Prozesse bedienen sich für diese Aufgaben des SMON.

Hauptaufgabe des PMON ist die Kontrolle aller Prozesse. Stürzt einer der Verwaltungsprozesse ab oder produziert einen Fehler, so werden die nun nicht mehr benötigten Speicher-Ressourcen durch den PMON freigegeben und ggf. der entsprechende Prozess nachgestartet.

## Hinweis

### Zur Erläuterung:

**Tablespace:** Die Daten einer Oracle-Datenbank werden in Tablespaces gespeichert. Ein Tablespace gehört immer zu genau einer Datenbank. Ein Tablespace kann sich über eine oder mehrere Datendateien (Betriebssystemdateien) erstrecken. In einem Tablespace können ein oder mehrere Segmente gespeichert werden. Segmente können z.B. Tabellen, Indizes oder andere DB-Objekte enthalten.

**Schema:** Ist eine logische Trennung von Datenbankobjekten. Jeder Oracle Nutzer hat grundsätzlich sein eigenes Schema, so dass UserX und UserY jeweils eine Tabelle "beispiel" anlegen können, ohne dass es zu Namens- oder Zugriffskonflikten kommt. Jeder User hat per Default nur Zugriff auf sein eigenes Schema, jede weitergehende Zugriffsberechtigung muss per GRANT explizit erteilt werden.

**Segment:** Jedes Datenbankobjekt besteht aus mindestens einem Segment, das genau in einem Tablespace liegt. Beim Anlegen eines Objekts allokiert die Datenbank MINEXTENTS Extents und fasst diese zu einem Segment zusammen. Bei Wachstum eines solchen Objekts allokiert die Datenbank weitere Extents, bis MAXEXTENTS erreicht ist.

**Extent:** Ein Extent besteht aus mehreren Datenblöcken und stellt einen zusammenhängenden Speicherbereich dar. Oracle allokiert Speicher für einzelne Segmente immer in Extent-Einheiten. Größe und Anzahl von Extents werden bei den jeweiligen Segmenten über die Storage-Parameter der Objekte eingestellt.

**Oracle-Block:** Ein Datenblock ist die kleinste physische Einheit, die Oracle zum Speichern von Daten verwendet. Der Wert wird bestimmt über den Initialisierungsparameter DB\_BLOCK\_SIZE (2 KB bis 32 KB). Bis Oracle 8i kann nur eine Blockgröße für die gesamte Datenbank definiert werden (dies geschieht beim Erzeugen der Datenbank). Seit Oracle 9i können bis zu fünf verschiedene Blockgrößen definiert werden.

#### **Die logische & physische Speicherhierarchie einer Oracle-Datenbank:**

Instanz → Datenbank → Tablespace → Segment → Extent → Oracle-Block