

ORA9A

Autor: Elmar Fuchs

Inhaltliches Lektorat: Leo Schütt-Cardaun

1. Auflage vom 24. November 2005

© by HERDT-Verlag für Bildungsmedien GmbH,  
Bodenheim

Internet: [www.herdt4you.de/.at/.ch](http://www.herdt4you.de/.at/.ch)  
[www.herdt4business.de/.at/.ch](http://www.herdt4business.de/.at/.ch)  
[www.herdt4vhs.de/.at](http://www.herdt4vhs.de/.at)

Alle Rechte vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Microfilm oder einem anderen Verfahren) ohne schriftliche Genehmigung des Herausgebers reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Diese Unterlage wurde mit großer Sorgfalt erstellt und geprüft. Trotzdem können Fehler nicht vollkommen ausgeschlossen werden. Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Die Bildungsmedien des HERDT-Verlags enthalten Links bzw. Verweise auf Internetseiten anderer Anbieter. Auf Inhalt und Gestaltung dieser Angebote hat der HERDT-Verlag keinerlei Einfluss. Hierfür sind alleine die jeweiligen Anbieter verantwortlich.

## Oracle 9i

Architektur und Administration

ORA9A

<b>1 Einführung .....</b>	<b>6</b>	<b>6 Werkzeuge für die Datenbankverwaltung .....</b>	<b>52</b>
1.1 Voraussetzungen und Ziele.....	6	6.1 SQL*Plus .....	52
1.2 Aufbau und Konventionen.....	7	6.2 Der Oracle Enterprise Manager .....	57
1.3 Die Tätigkeit eines Datenbankadministrators.....	8	6.3 SQL*Plus/OEM.....	60
<b>2 Basisarchitektur .....</b>	<b>10</b>	6.4 Schnellübersicht .....	61
2.1 Die Datenbankarchitektur im Überblick .....	10	6.5 Übung .....	61
2.2 Der Aufbau der Datenbankinstanz .....	11	<b>7 Verwalten der Datenbank.....</b>	<b>62</b>
2.2.1 Aufgaben der Datenbankinstanz.....	11	7.1 Initialisierungsdateien erstellen und verwalten .....	62
2.2.2 Speicher-Strukturen.....	12	7.2 Start und Stopp der Datenbank.....	66
2.2.3 Prozess-Strukturen .....	15	7.3 Überwachung der Datenbank .....	71
2.3 Der Aufbau der Datenbank .....	17	7.4 Systeminformationen im Data Dictionary .....	73
2.4 Weitere Dateien für den Datenbankbetrieb ....	20	7.5 Sessionverwaltung.....	75
2.5 Schnellübersicht.....	20	7.6 Schnellübersicht .....	80
2.6 Übung .....	21	7.7 Übung .....	80
<b>3 Arbeitsweise der Datenbank.....</b>	<b>22</b>	<b>8 Systemdateien und Instanzbetrieb .....</b>	<b>82</b>
3.1 Abarbeitung von SQL-Anweisungen .....	22	8.1 Control-Dateien.....	82
3.2 Transaktionskonzept .....	23	8.2 Verwalten der Control-Dateien.....	85
3.3 Schnellübersicht.....	27	8.3 Redo-Log-Dateien .....	86
3.4 Übung .....	27	8.4 Verwalten der Redo-Log-Dateien.....	89
<b>4 Installation der Datenbank-Software.....</b>	<b>28</b>	8.5 Archivierung der Redo-Log-Dateien.....	92
4.1 Voraussetzungen .....	28	8.6 Schnellübersicht .....	94
4.2 Die Optimal Flexible Architektur (OFA) .....	30	8.7 Übung .....	95
4.3 Die interaktive Installation mit dem Oracle Universal Installer .....	31	<b>9 Tablespaces und Datenbankdateien.....</b>	<b>96</b>
4.4 Installation mit einer Konfigurationsdatei.....	34	9.1 Tablespaces .....	96
4.5 Schnellübersicht.....	35	9.2 Verwalten von Tablespaces .....	99
4.6 Übung .....	35	9.3 Undo-Tablespace .....	104
<b>5 Erstellen der Datenbank.....</b>	<b>36</b>	9.4 Temporary Tablespace .....	107
5.1 Voraussetzungen .....	36	9.5 Übung .....	108
5.2 National Language Support .....	37	<b>10 Speicherung und Verwaltung der Daten .....</b>	<b>110</b>
5.3 Erstellen der Datenbank mit dem Oracle Database Configuration Assistant (ODCA) ....	38	10.1 Die logische Struktur der Datenspeicherung.....	110
5.4 Manuelle Erstellung der Datenbank .....	43	10.2 Tabellen.....	114
5.5 Konfiguration der Administrationsbenutzer.....	48	10.3 Index-Strukturen .....	120
5.6 Externe Authentifizierung .....	48	10.4 Constraints.....	123
5.7 Oracle Managed Files .....	50	10.5 Übung .....	127
5.8 Schnellübersicht.....	50		
5.9 Übung .....	51		

<b>11 Sicherheit .....</b>	<b>128</b>
11.1 Profile .....	128
11.2 Benutzerverwaltung .....	131
11.3 Privilegien .....	135
11.4 Rollen .....	139
11.5 Überwachung .....	142
11.6 Übung .....	145
<b>12 Oracle Net .....</b>	<b>146</b>
12.1 Netz-Architektur .....	146
12.2 Konfigurations-Dateien .....	148
12.3 Netzkonfiguration mit dem Oracle Net Manager .....	151
12.4 Die Steuerung des Listeners .....	153
12.5 Shared Server .....	155
12.6 Übung .....	156
<b>13 Backup und Recovery .....</b>	<b>158</b>
13.1 Sicherungsstrategien .....	158
13.2 Logische Online-Sicherung - Export und Import von Daten .....	159
13.3 Offline-Sicherung der Dateien .....	163
13.4 Manuelles Backup im Online-Modus .....	164
13.5 Wiederherstellung der Daten .....	165
13.6 Übung .....	167
<b>A1 Antworten auf Übungsfragen .....</b>	<b>168</b>
<b>A2 Deinstallation der Datenbank-     Software .....</b>	<b>182</b>
<b>Stichwortverzeichnis .....</b>	<b>184</b>

## 3 Arbeitsweise der Datenbank

### In diesem Kapitel erfahren Sie

- welche Prozesse beim Aufruf einer SQL-Anweisung abgearbeitet werden
- welche Bedeutung Transaktionen, Sperrungen und die Lesekonsistenz für die Arbeit mit der Datenbank haben

### Voraussetzungen

- ✓ Kenntnisse der Oracle Datenbankarchitektur

## 3.1 Abarbeitung von SQL-Anweisungen

### Abfragen

Die Durchführung einer SQL-Anweisung ist ein komplexer Vorgang, an dem nahezu alle Komponenten der Instanz beteiligt sind.

Ein SQL-Befehl `SELECT` wird nach der Übergabe vom Client-Programm an einen Server-Prozess von diesem ausgeführt. Dies umfasst mehrere Schritte:

Parse	<p>Der SQL-Befehl wird analysiert und aufbereitet.</p> <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Suche nach einer identischen Anweisung im Library Cache</li> <li><input checked="" type="checkbox"/> Syntax-Check des SQL-Quellcodes</li> <li><input checked="" type="checkbox"/> Semantische Prüfung der Tabellen- und Feldnamen im SQL über Systeminformationen im Dictionary Cache</li> <li><input checked="" type="checkbox"/> Prüfung der Rechte des Benutzers zur Durchführung des Befehls</li> <li><input checked="" type="checkbox"/> Erstellen eines lauffähigen Kommandos mit dem Ausführungsplan des Datenzugangs</li> <li><input checked="" type="checkbox"/> Speichern der SQL-Anweisung im Library Cache</li> </ul>
Bind	Vorhandenen Variablen werden die Werte zugewiesen.
Execute	<p>Der Prozess wird ausgeführt.</p> <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Wurden die Daten bereits verwendet und befinden sie sich im Database Buffer Cache, werden sie aus diesem ausgelesen.</li> <li><input checked="" type="checkbox"/> Befinden sich die Daten nicht im Buffer, werden sie aus der Datenbank ausgelesen und im Buffer hinterlegt.</li> </ul>
Fetch	Die Ergebnismenge wird an den Benutzer-Prozess übergeben.

### Datenmanipulation

Bei der Ausführung eines DML-Befehls (Data Manipulation Language, ein Teilbereich von SQL) wie `INSERT`, `UPDATE` oder `DELETE` sind die Phasen Parse und Bind identisch zum Befehl `SELECT`. Die Fetch-Phase entfällt, da keine Daten zurückgegeben werden. Der Inhalt der Execute-Phase unterscheidet sich von einem Befehl `SELECT`, da die Änderungen an den Daten protokolliert werden müssen.

<b>Execute</b>	<ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Befinden sich die Daten nicht im Buffer, werden sie aus der Datenbank ausgelesen und im Buffer hinterlegt.</li> <li><input checked="" type="checkbox"/> Die zu ändernden Datensätze werden gesperrt.</li> <li><input checked="" type="checkbox"/> In den Undo-Blöcken wird der alte Status der Daten aufgezeichnet, sodass die Aktion rückgängig gemacht werden kann.</li> <li><input checked="" type="checkbox"/> Die neuen Daten werden in die Daten-Blöcke geschrieben.</li> <li><input checked="" type="checkbox"/> Beide Änderungen werden im Database Buffer Cache vorgenommen.</li> <li><input checked="" type="checkbox"/> Die Blöcke werden als Dirty Buffers gekennzeichnet. Dies bedeutet, dass die vorgenommenen Änderungen noch nicht in die Daten-Dateien geschrieben wurden.</li> <li><input checked="" type="checkbox"/> Alle Manipulationen werden als Redo protokolliert.</li> </ul>
----------------	---

### Veränderung der Datenstrukturen

Bei der Ausführung von DDL-Befehlen (Data Definition Language, ein Teilbereich von SQL) wird der Data Dictionary Bereich in die Ausführung einbezogen. In diesem Teil der SGA werden Informationen über die Datenbank gespeichert. Entsprechend beziehen sich einige Aktivitäten in den Phasen Parse und Execute auf diesen Bereich, da die DDL-Befehle DML-Befehle in den Systemtabellen erfordern.

## 3.2 Transaktionskonzept

### Transaktionen

Datenbanken müssen beim laufenden Betrieb mit auftretenden Fehlern sowie mit parallelen Zugriffen mehrerer Benutzer umgehen können. Diesen Anforderungen wird durch das Transaktionskonzept entsprochen.

Die Transaktion besteht aus einem oder mehreren SQL-Befehlen, die als eine logische Einheit ausgeführt werden. Kann auf Grund eines Fehlers eine Operation nicht ausgeführt werden, wird keine der Operationen der Transaktion ausgeführt. Die Datenbank befindet sich sowohl am Anfang als auch am Ende der Transaktion in einem konsistenten Zustand. Werden mehrere dieser Befehlsfolgen gleichzeitig abgearbeitet, muss jede von ihnen vollständig isoliert ablaufen.

Ein typisches Beispiel ist eine Kontenbuchung. Von einem Konto A wird im 1. Schritt ein Betrag abgebucht, um diesen im 2. Schritt dem Konto B zuzuführen. Wird jedoch nur der 1. Schritt der Kontenbewegung ausgeführt, z. B. durch einen Systemabsturz, ist die Summe der Kontenbestände nicht mehr korrekt. In diesem Fall muss die Abbuchung von Konto A zurückgenommen werden.

<i>Notizen</i>	
----------------	--

## Das ACID-Prinzip

Das Datenbanksystem garantiert bei der Ausführung einer Transaktion die Einhaltung des ACID-Prinzips. Dieses beinhaltet vier grundlegende Eigenschaften:

<b>Atomicity - Atomarität</b> 'Alles oder Nichts'	Die Transaktion, die aus einer oder mehreren Operationen bestehen kann, wird entweder vollständig ausgeführt oder gar nicht. Tritt während der Transaktion ein Fehler auf (z. B. ein Programmfehler, Hardwarefehler oder ein Betriebssystemabsturz), werden alle bisherigen Operationen der Transaktion rückgängig gemacht.
<b>Consistency - Konsistenz</b>	Ist die Transaktion abgeschlossen, befindet sich die Datenbank in einem konsistenten Zustand. Dazu tragen die definierten Integritätsbedingungen (Wertebereiche, Schlüsseleigenschaften, Fremdschlüssel usw.) bei, welche die logische Konsistenz sichern. Verletzungen dieser Integritätsbedingungen führen zum Zurücksetzen der Transaktion.
<b>Isolation - Isolation</b>	Transaktionen laufen isoliert ab, mehrere gleichzeitig ablaufende Transaktionen stören und beeinträchtigen sich gegenseitig nicht. Dies wird durch das DBMS durch Synchronisationsmaßnahmen erreicht. Neue Transaktionen erhalten immer nur gesicherte, konsistente Daten.
<b>Durability - Dauerhaftigkeit</b>	Das Ergebnis einer erfolgreichen Transaktion ist dauerhaft. Wird eine Transaktion als erfolgreich abgeschlossen gemeldet, können die - gegebenenfalls - geänderten Daten nicht mehr durch Fehler jeglicher Art verloren gehen. Dies gilt auch für den Fall, dass sich die Daten noch im Speicher befinden und noch nicht in die Daten-Dateien geschrieben wurden.

## Transaction Isolation Level

Durch den Transaction Isolation Level wird der Grad der Parallelität von Transaktionen gesteuert. Je höher Sie den Level wählen, desto weniger Probleme mit der Konsistenz treten auf. Gleichzeitig verringert sich aber mit einem höheren Level auch der Datendurchsatz und damit die Performance der Datenbank.

Bei der gleichzeitigen Ausführung mehrerer Transaktionen können verschiedene typische Probleme auftreten.

- Dirty Reads - Eine Transaktion liest Daten, die von einer anderen Transaktion geschrieben wurden, die jedoch zu einem späteren Zeitpunkt abgebrochen wird. Die gelesenen Daten sind damit ungültig, da sie entsprechend der Atomaritätseigenschaft zurückgesetzt werden.
- Nonrepeatable Reads - Eine Transaktion will die gleichen Daten wiederholt lesen, diese Daten wurden jedoch zwischenzeitlich von einer anderen Transaktion verändert.
- Phantom Read - Eine Transaktion führt eine Abfrage durch und erhält eine Auswahl von Daten. Eine andere Transaktion fügt dem Datenbestand weitere Daten hinzu bzw. entfernt sie. Die gleiche Leseoperation der ersten Transaktion würde nun eine andere Auswahl von Daten als Ergebnis erhalten.

Von diesen Problemen ausgehend wurden im Standard SQL:1999 (SQL2) die Transaction Isolation Level einer Transaktion definiert.

Die folgende Tabelle zeigt die bei den verschiedenen Transaction Isolation Levels möglichen Konkurrenz-situationen:

Transaction Isolation Level	Beschreibung	Dirty Reads	Nonrepeatable Reads	Phantom Read
Read Uncommitted	Auf Daten, die von einer Transaktion geschrieben werden, kann mittels einer anderen Transaktion lesend zugegriffen werden.	Ja	Ja	Ja
Read Committed	Datensätze, die von einer Transaktion modifiziert werden, sind gesperrt, Datensätze, die nur gelesen werden, jedoch nicht.	Nein	Ja	Ja



### Beispiel: Prinzip der Ausführung eines DML-Befehls

Die Anweisung wird durch einen Server-Prozess im Dedicated Modus ausgeführt. Da ein DML-Befehl Daten ändert, muss der Befehl im Rahmen einer Transaktion bzw. als eigenständige Transaktion ausgeführt werden.

- ⇒ Als Erstes überprüft der Server Prozess, ob ein identischer Befehl, der die gleichen Objekte manipuliert, schon im Library Cache vorliegt. Ist dies der Fall und sind die Zugriffsrechte korrekt, wird dieser Befehl verwendet ①.
- ⇒ Liegt der Befehl nicht vor, werden die Zugriffsrechte geprüft, der Befehl wird übersetzt und im Library Cache gespeichert. Der Server-Prozess benötigt dazu Systeminformationen. Diese findet er im Dictionary Cache, bzw. er muss sie aus der Datenbank auslesen und im Dictionary Cache speichern ②.
- ⇒ Die Ausführung des Befehls beginnt mit der Suche nach den benötigten Daten. Sie werden zuerst im Database Buffer gesucht ③.
- ⇒ Falls die Daten dort nicht vorliegen, werden sie aus der Datenbank gelesen ④ und im Database Buffer gespeichert ⑤.
- ⇒ Reicht der freie Platz im Database Buffer zur Speicherung nicht aus, erhält der DBWR-Prozess den Auftrag, diesen herzustellen. Er gibt belegte Blöcke für neue Inhalte frei. Bei Blöcken mit manipulierten Daten (Dirty Blocks) müssen diese zuerst vom DBWR in die Datenbank geschrieben werden ⑥.
- ⇒ Vor der Veränderung der Daten im Database Buffer werden die ursprünglichen Werte zusammen mit einem Verweis auf die Transaktion in dem Undo-Segment gesichert.
- ⇒ Alle Änderungen - sowohl die an den Datenblöcken als auch jene an den Undo-Segmentblöcken - protokolliert der Server-Prozess im Redo Log Buffer ⑦.
- ⇒ Wird die Transaktion mit einem COMMIT abgeschlossen, schreibt der LGWR-Prozess die Protokolle des Redo Log Buffers in den Redo-Log-Dateien fort ⑧. Die Aktion ist erst dann erfolgreich abgeschlossen, wenn die Daten in den Redo-Log-Dateien auf der Festplatte geschrieben sind.

Es müssen bei einem Commit keine Datenblöcke in die Daten-Dateien geschrieben werden.





