

# Auf einen Blick

1	Einführung .....	23
2	Verwendete Werkzeuge und Ressourcen .....	35
TEIL I Grundlagen		
3	Aufbau der Datenbank aus Sicht eines Programmierers .....	57
4	Datenbankobjekte und SQL .....	113
5	Datensicherheit, -konsistenz und Transaktion .....	159
6	Programmierung der Datenbank .....	213
TEIL II Die Sprache PL/SQL		
7	Die Blockstruktur und Syntax von PL/SQL .....	233
8	Kontrollstrukturen .....	275
9	Datentypen in PL/SQL .....	303
10	Cursor .....	341
11	Events in der Datenbank: Programmierung von Triggern .....	385
12	Packages .....	449
13	Erweiterung von SQL .....	521
14	Dynamisches SQL .....	571
15	Exception .....	631
TEIL III PL/SQL im Einsatz		
16	Arbeiten mit LOBs (Large Objects) .....	667
17	Arbeiten mit XML .....	701
18	Arbeiten mit JSON .....	769
19	Objektorientierung .....	801
20	Integration von Oracle in Applikationen .....	881
21	Performance-Tuning und Codeanalyse .....	945
22	Workshop: PL/SQL Instrumentation Toolkit (PIT) .....	1003

# Inhalt

Materialien zum Buch .....	21
<b>1 Einführung</b> .....	<b>23</b>
<b>1.1 Für wen ist dieses Buch geschrieben?</b> .....	<b>23</b>
<b>1.2 Der Aufbau des Buches</b> .....	<b>26</b>
1.2.1 Teil I: Grundlagen .....	27
1.2.2 Teil II: Die Sprache PL/SQL .....	28
1.2.3 Teil III: PL/SQL im Einsatz .....	31
<b>1.3 Vorwort zur vierten Auflage</b> .....	<b>34</b>
<b>2 Verwendete Werkzeuge und Ressourcen</b> .....	<b>35</b>
<b>2.1 Oracles Online-Dokumentation</b> .....	<b>35</b>
2.1.1 Wo finde ich die benötigten Informationen? .....	36
2.1.2 PL/SQL-Grundlagen .....	37
2.1.3 Oracle-Packages .....	38
2.1.4 Weiterführende Literatur .....	39
<b>2.2 Aufsetzen einer Beispieldatenbank</b> .....	<b>40</b>
<b>2.3 SQL*Plus</b> .....	<b>41</b>
<b>2.4 SQLCL</b> .....	<b>42</b>
<b>2.5 SQL Developer</b> .....	<b>43</b>
<b>2.6 »explain plan«</b> .....	<b>44</b>
<b>2.7 Autotrace</b> .....	<b>46</b>
<b>2.8 Runstats</b> .....	<b>48</b>
<b>2.9 Trace und tkprof</b> .....	<b>49</b>
<b>2.10 Debugger</b> .....	<b>52</b>
<b>2.11 Weitere Werkzeuge</b> .....	<b>53</b>
<b>2.12 Die Beispielskripte</b> .....	<b>53</b>

TEIL I Grundlagen

3 Aufbau der Datenbank aus Sicht eines Programmierers 57

---

3.1 Grundlegende Arbeitsweise der Datenbank ..... 57

3.1.1 Anforderungen an ein Datenbankmanagementsystem ..... 58

3.1.2 Lesekonsistenz ..... 61

3.1.3 Die Begriffe »Datenbank«, »Schema« und »Tablespace« ..... 62

3.1.4 Systemtabellen, Data Dictionary und Privilegien ..... 64

3.1.5 Die Sicht der Anwendung auf die Datenbank ..... 66

3.2 Logischer Aufbau: Schema, Tablespace und Co. .... 69

3.2.1 Schema ..... 69

3.2.2 Tablespace ..... 73

3.2.3 Auswirkungen auf die Architektur einer Applikation ..... 76

3.3 Die physikalische Datenbank ..... 79

3.3.1 Datendateien ..... 79

3.3.2 Redo-Log-Dateien ..... 81

3.3.3 Kontrolldatei ..... 82

3.3.4 Parameterdatei ..... 82

3.3.5 Passwortdatei ..... 83

3.4 Instanz und Speicherstrukturen ..... 84

3.4.1 Die Speicherbereiche der SGA ..... 86

3.4.2 Shared Pool ..... 88

3.4.3 Die Hintergrundprozesse ..... 89

3.5 Containerdatenbank ..... 94

3.6 Start der Datenbank ..... 95

3.7 Verbindungsaufbau zur Datenbank ..... 96

3.7.1 Verbindungsarten und Treiber ..... 98

3.7.2 Dedicated-Server-Verbindung ..... 104

3.7.3 Shared-Server-Verbindung ..... 105

3.7.4 Database Resident Connection Pool ..... 107

3.7.5 Und nun? Entscheidungshilfen für den Verbindungsaufbau ..... 109

## 4 Datenbankobjekte und SQL 113

---

<b>4.1</b>	<b>Tabellen</b> .....	113
4.1.1	Heap Organized Table .....	113
4.1.2	Index Organized Table .....	115
4.1.3	Temporäre Tabellen .....	116
4.1.4	Partitionierte Tabellen .....	118
<b>4.2</b>	<b>Index</b> .....	120
4.2.1	Anmerkung zur Benutzung von Indizes .....	122
4.2.2	B*-Baum-Index .....	124
4.2.3	Reverse-Key-Index .....	126
4.2.4	Funktionsbasierter Index .....	127
<b>4.3</b>	<b>Views und Materialized Views</b> .....	129
4.3.1	Views .....	129
4.3.2	Materialized Views .....	130
<b>4.4</b>	<b>PL/SQL-Programm</b> .....	132
<b>4.5</b>	<b>Sonstige Datenbankobjekte</b> .....	133
4.5.1	Sequenzen .....	133
4.5.2	Synonym .....	134
4.5.3	Datenbanklink .....	135
4.5.4	Große Datenmengen: »CLOB«, »NCLOB«, »BLOB« und »BFile« .....	136
4.5.5	Benutzerdefinierte Typen, XML, JSON .....	138
4.5.6	Weitere Datenbankobjekte .....	139
<b>4.6</b>	<b>Exkurs: Zeichensatzcodierung</b> .....	139
4.6.1	Zeichensatzcodierung im Überblick .....	139
4.6.2	Zeichensatzcodierung bei Oracle .....	141
<b>4.7</b>	<b>Mächtigkeit von SQL</b> .....	145
4.7.1	Analytische Funktionen .....	145
4.7.2	Hierarchische Abfragen .....	148
4.7.3	Error-Logging .....	151
4.7.4	Fazit .....	156

## 5 Datensicherheit, -konsistenz und Transaktion 159

---

<b>5.1</b>	<b>Lese- und Schreibkonsistenz</b> .....	160
5.1.1	Lesekonsistenz .....	160
5.1.2	Schreibkonsistenz .....	164

<b>5.2</b>	<b>Transaktion</b>	164
5.2.1	Transaktion zum Schutz der Lesekonsistenz	164
5.2.2	Transaktion zur Definition eines Geschäftsfalls	166
5.2.3	Zusammenfassung	167
<b>5.3</b>	<b>Datenkonsistenz und referenzielle Integrität</b>	168
5.3.1	Datenintegrität	169
5.3.2	Performance-Überlegungen zu Datenbank-Constraints	176
5.3.3	Datenkonsistenz	179
5.3.4	Zusammenfassung	183
<b>5.4</b>	<b>Explizites Sperren von Daten durch die Anwendung</b>	183
5.4.1	Das Problem: Lost Updates	183
5.4.2	Das optimistische Sperren	184
5.4.3	Das pessimistische Sperren	188
5.4.4	Das vorsichtig optimistische Sperren	188
5.4.5	Und nun? Wann sollten Sie welche Sperrstrategie verwenden?	189
<b>5.5</b>	<b>Verarbeitung einer SQL-Anweisung</b>	190
5.5.1	Parsen und Optimierung	191
5.5.2	Datenlieferung über Cursor	196
<b>5.6</b>	<b>Die Sperrmechanismen von Oracle</b>	196
5.6.1	Locks	196
5.6.2	Latches	197
<b>5.7</b>	<b>Datensicherheit</b>	197
<b>5.8</b>	<b>Workshop: Einfluss der Programmierung</b>	200
5.8.1	Das Ziel unserer Programmierung	200
5.8.2	Implementierung des Tests	202

---

## 6 Programmierung der Datenbank 213

<b>6.1</b>	<b>Erweiterung der Datenbankfunktionalität</b>	213
<b>6.2</b>	<b>Programmierung der Datenkonsistenz</b>	215
6.2.1	Datenbanktrigger	216
6.2.2	Datenzugriff über PL/SQL	220
6.2.3	Datenkonsistenz jenseits referenzieller Integrität	222
<b>6.3</b>	<b>Programmierung der Datensicherheit</b>	223

<b>6.4</b>	<b>Anwendungsprogrammierung mit PL/SQL .....</b>	<b>226</b>
6.4.1	PL/SQL auf der Client-Seite .....	227
<b>6.5</b>	<b>Unterstützung der Administration durch PL/SQL .....</b>	<b>227</b>
6.5.1	Einsatz von PL/SQL in Skripten .....	228
6.5.2	Verwaltung wiederkehrender Aufgaben mit Scheduler und Jobs .....	229
6.5.3	Datenbanktrigger im Umfeld der Datensicherung und des Auditings .....	229

## TEIL II Die Sprache PL/SQL

### **7 Die Blockstruktur und Syntax von PL/SQL** 233

---

<b>7.1</b>	<b>Das Grundgerüst: der PL/SQL-Block .....</b>	<b>234</b>
7.1.1	Deklaration von Variablen .....	238
7.1.2	Schachtelung von Blöcken zur Fehlerbehandlung .....	240
7.1.3	Gültigkeitsbereich von Variablen .....	241
<b>7.2</b>	<b>Prozeduren .....</b>	<b>241</b>
7.2.1	Prozeduren mit Parametern .....	245
7.2.2	Formen der Parameterzuweisung .....	251
7.2.3	Optionale Parameter .....	252
7.2.4	Beliebig viele Parameter an eine Methode übergeben .....	256
<b>7.3</b>	<b>Funktionen .....</b>	<b>257</b>
<b>7.4</b>	<b>Datenbanktrigger .....</b>	<b>261</b>
<b>7.5</b>	<b>Packages .....</b>	<b>262</b>
7.5.1	Package-Spezifikation .....	263
7.5.2	Package-Körper .....	264
7.5.3	Aufruf von Prozeduren und Methoden in Packages .....	267
<b>7.6</b>	<b>Ausführungsrechte von PL/SQL-Blöcken .....</b>	<b>267</b>
<b>7.7</b>	<b>Compiler-Anweisungen (Pragma) .....</b>	<b>270</b>
7.7.1	Die autonome Transaktion .....	271
7.7.2	Initialisierung eigener Fehler .....	272
<b>7.8</b>	<b>Best Practices .....</b>	<b>272</b>

8.1	Auswertende Anweisung 1 (»if then else«-Anweisung)	275
8.2	Auswertende Anweisung 2 (»case«-Anweisung)	277
8.2.1	Einfache »case«-Anweisung	277
8.2.2	Aufruf der »case«-Anweisung als SQL-Ausdruck	278
8.2.3	Die auswertende »case«-Anweisung	279
8.3	Einfache Schleifen	281
8.3.1	Basisschleife (Schleife)	281
8.3.2	Abweisende Schleife 1 (»for«-Schleife)	283
8.3.3	Abweisende Schleife 2 (»while«-Schleife)	288
8.3.4	Best Practices	289
8.4	Konditionale Kompilierung	292
8.4.1	Die Auswahldirektive (Selection Directive)	293
8.4.2	Die Abfragedirektive (Inquiry Directive)	294
8.4.3	Die Errordirektive (Error Directive)	297
8.5	Aus der Mottenkiste: Konzepte, die Sie nicht verwenden sollten	297
8.5.1	Label	298
8.5.2	»continue«- und »goto«-Anweisung	300

9.1	Skalare Datentypen	303
9.1.1	SQL-Datentypen	303
9.1.2	Abweichende Größen von PL/SQL-Datentypen	306
9.1.3	Basistypen und Untertypen in PL/SQL	307
9.1.4	SQL-Datentypen, die in PL/SQL nicht existieren	309
9.1.5	PL/SQL-Datentypen, die in SQL nicht existieren	309
9.1.6	Objektorientierte Typen	309
9.1.7	Benutzerdefinierte Datentypen	311
9.1.8	Ableitung von Variablentypen aus dem Data Dictionary	311
9.2	Kollektionen in PL/SQL	314
9.2.1	Record	314
9.2.2	Assoziative Tabellen	326
9.2.3	Massenverarbeitung mit assoziativen Tabellen	330
9.2.4	»VARRAY« oder »NESTED_TABLE« als Alternative zu einer assoziativen Tabelle	339
9.3	Cursor	340

<b>10</b>	<b>Cursor</b>	341
<hr/>		
<b>10.1</b>	<b>Lebenszyklus eines Cursors</b>	341
10.1.1	Deklaration eines Cursors	341
10.1.2	Lesen eines Datensatzes aus dem Cursor	342
10.1.3	Schließen des Cursors	343
<b>10.2</b>	<b>Cursor-Attribute</b>	344
<b>10.3</b>	<b>Parametrisierte Cursor</b>	347
<b>10.4</b>	<b>Mengenverarbeitung mit »bulk collect«</b>	349
<b>10.5</b>	<b>Kurzform: die »cursor for«-Schleife</b>	351
<b>10.6</b>	<b>Implizite versus explizite Cursor</b>	353
10.6.1	Implizite oder explizite Cursor	353
10.6.2	Implizite oder explizite Cursor-Kontrolle	357
<b>10.7</b>	<b>Cursor-Variablen (»ref«-Cursor)</b>	359
10.7.1	Starke Cursor-Variable	360
10.7.2	Schwache Cursor-Variable	361
<b>10.8</b>	<b>Cursor-Ausdrücke</b>	366
<b>10.9</b>	<b>Gemeinsamer Zugriff auf Daten über verteilte Cursor</b>	371
<b>10.10</b>	<b>Tabellenfunktionen</b>	373
10.10.1	Was ist eine Tabellenfunktion?	373
10.10.2	Workshop: Tabellenfunktion	375
10.10.3	Verwendung von Tabellenfunktionen	379
10.10.4	Workshop: Erstellung einer Tabellenfunktion	380
10.10.5	Polymorphe Tabellenfunktion	383
<b>11</b>	<b>Events in der Datenbank: Programmierung von Triggern</b>	385
<hr/>		
<b>11.1</b>	<b>DML-Trigger</b>	385
11.1.1	Anweisungs- versus Zeilentrigger	386
11.1.2	Der Triggerkörper	391
11.1.3	Wann wird ein Trigger ausgelöst?	392
11.1.4	Das Mutating-Table-Problem	395
11.1.5	Compound Trigger	398
11.1.6	Workshop: Lösung des Mutating-Table-Problems mit einem Compound Trigger	399



11.1.7	Cross Edition Trigger .....	405
11.1.8	Benennungskonvention von Triggern .....	406
<b>11.2</b>	<b>»instead of«-Trigger .....</b>	<b>406</b>
<b>11.3</b>	<b>Einsatzbereiche von DML-Triggern .....</b>	<b>409</b>
11.3.1	Erweiterung der Datenkonsistenzprüfung über Constraints hinaus .....	409
11.3.2	Workshop: Statusänderungen in einer bestimmten Reihenfolge durchführen .....	410
11.3.3	Implementierung einfacher Geschäftsregeln .....	413
11.3.4	Historisierung, Logging und Auditing von Daten .....	414
11.3.5	Workshop: generisches Logging .....	415
11.3.6	Workshop: Historisierung von Daten mit einem »instead of«-Trigger .....	423
<b>11.4</b>	<b>Wann Sie DML-Trigger nicht verwenden sollten .....</b>	<b>429</b>
11.4.1	Auditing mithilfe von Triggern .....	431
11.4.2	Schutz der Datenintegrität .....	432
<b>11.5</b>	<b>Datenbanktrigger .....</b>	<b>433</b>
11.5.1	Ereignisattribute .....	434
11.5.2	Datenbankereignisse .....	437
11.5.3	Benutzerbezogene Ereignisse .....	439
11.5.4	DDL-Ereignisse .....	443
11.5.5	Systemereignisse .....	444
<b>11.6</b>	<b>Zusammenfassung .....</b>	<b>445</b>

## 12 Packages

449

---

<b>12.1</b>	<b>Trennung von öffentlicher und privater Logik .....</b>	<b>449</b>
12.1.1	Deklaration .....	450
12.1.2	Implementierung .....	452
12.1.3	Zusammenfassung .....	463
<b>12.2</b>	<b>Überladung in Packages .....</b>	<b>465</b>
12.2.1	Deklaration .....	465
12.2.2	Implementierung .....	467
12.2.3	Zusammenfassung .....	470
<b>12.3</b>	<b>Ausführungsrechte von Packages .....</b>	<b>471</b>
12.3.1	Rollen und Berechtigungskonzepte .....	473

12.3.2	Erweiterungen des Aufruferrechte-Berechtigungskonzepts .....	474
12.3.3	Steuerung des Zugriffs auf ein Package .....	475
<b>12.4</b>	<b>Packages und die Dependency Chain .....</b>	<b>476</b>
<b>12.5</b>	<b>Verschlüsselung von Package-Code .....</b>	<b>483</b>
12.5.1	Das »wrap«-Utility .....	484
12.5.2	Verwendung des Packages »dbms_ddl« .....	485
<b>12.6</b>	<b>Oracle-Packages .....</b>	<b>488</b>
12.6.1	Das Package »standard« .....	491
12.6.2	Wichtige Oracle-Packages .....	493
<b>12.7</b>	<b>Workshop: Verwaltung von Anwendungsparametern .....</b>	<b>501</b>
12.7.1	Das Problem und die Lösungsidee .....	501
12.7.2	Vorüberlegungen zur Parametertabelle .....	502
12.7.3	Die Parametertabelle .....	504
12.7.4	Einrichtung der Parametertabelle und der Zugriffsrechte .....	509
12.7.5	Das Parameter-Package .....	511
12.7.6	Das Package im Einsatz .....	517

## **13 Erweiterung von SQL** 521

---

<b>13.1</b>	<b>Wann SQL erweitert werden sollte .....</b>	<b>521</b>
13.1.1	Bleiben Sie auf dem aktuellen Wissensstand .....	522
13.1.2	Voraussetzungen für die Erweiterung von SQL .....	526
<b>13.2</b>	<b>SQL durch eigene Funktionen erweitern .....</b>	<b>527</b>
13.2.1	Anforderungen an den PL/SQL-Block .....	528
13.2.2	Nebenwirkungsfreiheit (Purity) .....	529
13.2.3	Optimizer Hints und Klauseln .....	529
13.2.4	Das Pragma »restrict_references« .....	530
13.2.5	Workshop: deterministische Funktion .....	530
<b>13.3</b>	<b>Workshop: Berechnung der Fakultät .....</b>	<b>537</b>
13.3.1	Einschränkung der Fakultätsfunktion auf definierte Werte .....	538
13.3.2	Zielvorgabe .....	540
13.3.3	Und was ist mit Rekursion? .....	545
<b>13.4</b>	<b>Gruppenfunktionen selbst erstellen .....</b>	<b>547</b>
13.4.1	Arbeitsweise von Gruppenfunktionen .....	548
13.4.2	Workshop: Erstellung einer Gruppenfunktion .....	551

13.4.3	Test der Gruppenfunktion .....	556
13.4.4	Zusammenfassung .....	558
<b>13.5</b>	<b>Workshop: Code-Generator für Gruppenfunktionen .....</b>	<b>558</b>

## **14    Dynamisches SQL** 571

---

<b>14.1</b>	<b>Dynamisches SQL mittels »execute immediate« .....</b>	<b>572</b>
14.1.1	Verwendung von Bindevariablen .....	573
<b>14.2</b>	<b>Dynamisches SQL mit Cursor-Variablen .....</b>	<b>576</b>
<b>14.3</b>	<b>Workshop: Erstellung einer Prozedur als Schnittstelle zu einem externen Programm .....</b>	<b>576</b>
14.3.1	Die Aufgabenstellung .....	576
14.3.2	Der Lösungsansatz .....	577
14.3.3	Vorbereitende Arbeiten .....	577
14.3.4	Die Prozedur für den Datenzugriff .....	578
<b>14.4</b>	<b>DBMS_SQL-Package .....</b>	<b>581</b>
14.4.1	Workshop: Code-Generator .....	584
<b>14.5</b>	<b>Sicherheit bei dynamischem SQL .....</b>	<b>597</b>
14.5.1	SQL-Injection über Suchparameter .....	597
14.5.2	SQL-Injection über Formatangaben .....	598
14.5.3	SQL-Injection über das Einschmuggeln zusätzlicher Anweisungen .....	599
14.5.4	Vermeidung von SQL-Injection 1: Bindevariablen .....	600
14.5.5	Vermeidung von SQL-Injection 2: »dbms_assert« .....	600
<b>14.6</b>	<b>SQL-Makros .....</b>	<b>601</b>
14.6.1	Skalare SQL-Makros .....	602
14.6.2	Tabellen-SQL-Makros .....	610
<b>14.7</b>	<b>Polymorphe Tabellenfunktionen .....</b>	<b>615</b>
14.7.1	Ein erstes Beispiel .....	616
14.7.2	Beispiel 2: Konvertierung in JSON oder XML .....	623
14.7.3	Mehrere PTF in einem Package .....	624
14.7.4	PTF zum Erzeugen neuer Zeilen .....	625
14.7.5	PTF und der Execution Store (XSTORE) .....	627

## 15 Exception 631

---

<b>15.1 Oracle-Fehler</b> .....	631
15.1.1 Benannte Fehler .....	635
15.1.2 sqlerrm- und sqlcode-Funktionen und der Fehler-Stack .....	637
15.1.3 Nicht benannte Fehler benennen .....	645
<b>15.2 Applikationsfehler erstellen und bearbeiten</b> .....	647
15.2.1 Fehler direkt mit »raise_application_error« erzeugen .....	647
15.2.2 Fehler aus einem Fehler-Package erstellen lassen .....	648
15.2.3 Zentralisierung der Fehlermeldungen über »lmsgen« .....	650
15.2.4 Workshop: Wrapper-Package um »utl_lms« .....	653
15.2.5 Zusammenfassung: Fehlermeldungen mit »utl_lms« .....	657
<b>15.3 Workshop: zentralisierter Fehler-Handler mit einem Trigger</b> .....	657
<b>15.4 Zusammenfassung</b> .....	662

## TEIL III PL/SQL im Einsatz

## 16 Arbeiten mit LOBs (Large Objects) 667

---

<b>16.1 Technische Struktur</b> .....	668
16.1.1 Einsatz von LOB-Datentypen in der Datenbank .....	668
16.1.2 LOB als PL/SQL-Variable .....	673
16.1.3 LOB als Methodenparameter .....	679
16.1.4 SecureFiles .....	681
<b>16.2 Die Datentypen »CLOB«, »NCLOB«, »BLOB« und »BFILE«</b> .....	683
16.2.1 CLOB und NCLOB .....	683
16.2.2 Der binäre Datentyp »BLOB« .....	684
16.2.3 BFile .....	684
<b>16.3 Das Package »DBMS_LOB«</b> .....	686
16.3.1 Schreibzugriff auf temporäre oder persistente LOBs .....	687
16.3.2 Verwaltung temporärer und persistenter LOBs .....	689
16.3.3 API für BFile-LOB .....	691
16.3.4 Zugriff auf LOBs durch die Anwendung .....	692
<b>16.4 Workshop: Hilfsfunktionen zum Arbeiten mit LOBs</b> .....	692

<b>17</b>	<b>Arbeiten mit XML</b>	701
<b>17.1</b>	<b>Der Datentyp »XMLType«</b>	701
17.1.1	Verwendung von »XMLType« als Tabellen- oder Spaltentyp	702
17.1.2	»XMLType«-Member-Functions	705
17.1.3	Umformung von XML mittels XSLT	707
<b>17.2</b>	<b>Die Speicherung von XML-Daten in der Datenbank</b>	710
<b>17.3</b>	<b>XML aus relationalen Daten erzeugen</b>	713
17.3.1	Der SQL/XML-Standard	714
17.3.2	Das Package »DBMS_XMLGEN«	718
<b>17.4</b>	<b>Relationale Daten aus XML extrahieren</b>	726
17.4.1	Extraktion relationaler Daten mit »XMLTable«	727
17.4.2	Extraktion relationaler Daten mittels Objektorientierung	730
<b>17.5</b>	<b>XML mit PL/SQL verarbeiten</b>	731
17.5.1	Die Programmierung mittels DOM-Baum	731
17.5.2	Die XML-Packages	733
<b>17.6</b>	<b>Die XML-Datenbank</b>	743
17.6.1	Einführung in die XML-Datenbank	744
17.6.2	Speicherung und Veröffentlichung binärer Dokumente und XML-Dokumente	747
17.6.3	Dokumente über XDB verwalten	750
17.6.4	Zugriffsschutz und Sicherheit von XDB	759
17.6.5	Versionierung von Ressourcen	765
<b>18</b>	<b>Arbeiten mit JSON</b>	769
<b>18.1</b>	<b>JSON</b>	769
18.1.1	Überblick: Was ist JSON?	769
18.1.2	Der Datentyp JSON	772
18.1.3	Der Datentyp JSON (revisited)	774
18.1.4	Abfragen gegen JSON-Instanzen	775
18.1.5	JSON Data Guide	777
<b>18.2</b>	<b>Programmierung von JSON mit PL/SQL</b>	780
18.2.1	Übersicht über die PL/SQL-JSON-Typen	781
18.2.2	Objektmethoden	782
18.2.3	Manipulation mittels APEX-Packages	785

<b>18.3</b>	<b>SODA (Simple Oracle Document Access)</b>	789
18.3.1	SODA-Kollektion	791
18.3.2	SODA-Operationen	794
18.3.3	SODA-Dokument	796
18.3.4	SODA und Transaktionen	797

---

<b>19</b>	<b>Objektorientierung</b>	801
-----------	---------------------------	-----

---

<b>19.1</b>	<b>Einführung in die Objektorientierung</b>	803
19.1.1	Alles ist ein Objekt	804
19.1.2	Das zweite Reizwort: Vererbung!	806
19.1.3	Abstrakte und finale Klassen	807
19.1.4	Statische Methoden	808
19.1.5	Objektidentität versus »Statement of Truth«	808
19.1.6	Klassen haben komplexe Strukturen	810
19.1.7	Auswirkungen auf die Datenbankprogrammierung	812
<b>19.2</b>	<b>Objektorientierte Datentypen</b>	815
19.2.1	»object«	815
19.2.2	»varray«	816
19.2.3	»nested table«	819
19.2.4	Vergleiche von Kollektionen	820
19.2.5	Methoden von Kollektionstypen	821
19.2.6	Workshop: Liste von Werten übergeben	823
<b>19.3</b>	<b>Objektorientierte Datenmodelle</b>	826
<b>19.4</b>	<b>Workshop: der Datentyp »MoneyType«</b>	830
19.4.1	Vorüberlegungen	830
19.4.2	Implementierung des Typs »MoneyType«	831
19.4.3	Der Typkörper	833
19.4.4	Implementierung des Packages »moneytype_pkg«	836
19.4.5	Der Package-Körper	837
19.4.6	Die Rechtesituation ab Version 11g	846
19.4.7	Erweiterung durch Vererbung	849
<b>19.5</b>	<b>Objektorientierte Anwendungsentwicklung und relationale Datenbanken</b>	851
19.5.1	Das Problem des Impedance Mismatch	852
19.5.2	Lösungsansatz 1: die Vision der generischen Datenbank	862
19.5.3	Lösungsansatz 2: objektrelationale Mapping-Werkzeuge	869
19.5.4	Lösungsansatz 3: das »Smart-Database«-Paradigma	872

20

Integration von Oracle in Applikationen

881

---

<b>20.1</b>	<b>Sperrung von Daten bei der Datenänderung</b>	882
20.1.1	Transaktionsschutz innerhalb der Datenbank	882
20.1.2	Erweiterter Fokus: Datensicherung im Umfeld von Anwendungen	883
20.1.3	Pessimistisches Locking	884
20.1.4	Optimistisches Sperren	889
20.1.5	Kombination aus optimistischem und pessimistischem Sperren	897
20.1.6	Database-Change-Notification-basiertes Locking	899
<b>20.2</b>	<b>Speicherung von Session-Informationen</b>	901
20.2.1	Grundlagen eines Kontextes	902
20.2.2	Session-Kontext	903
20.2.3	Globally Accessed Context	905
20.2.4	Workshop: Package zur Verwaltung von Kontexten	908
<b>20.3</b>	<b>Zugriff auf Daten über PL/SQL-Packages</b>	920
20.3.1	Kapselung von DML-Operationen in Packages	921
20.3.2	Vermeidung von Triggern durch Packages	922
20.3.3	Integration datenbezogener Geschäftsregeln	923
<b>20.4</b>	<b>Workshop: Keimzelle einer sicheren Datenbankanwendung</b>	924
20.4.1	Das Projekt	924
20.4.2	Übersicht über die Architektur	924
20.4.3	Die »logon«-Prozedur	927
20.4.4	Aufsetzen der Schemata	928
20.4.5	Die Packages	934
20.4.6	Test der Architektur	940
20.4.7	Zusammenfassung und Ausblick	942

21

Performance-Tuning und Codeanalyse

945

---

<b>21.1</b>	<b>Regeln zur Performance-Optimierung</b>	946
21.1.1	Nutzen Sie SQL, falls dies möglich ist	946
21.1.2	Betrachten Sie die Datenbank als entfernte Ressource	949
21.1.3	Benutzen Sie Bindevariablen	949
21.1.4	Arbeiten Sie mengenorientiert	950
21.1.5	Bereiten Sie die Daten vor der Programmierung optimal vor	951
21.1.6	Arbeiten Sie sich in die Grundkonzepte der Datenbank ein	951

21.1.7	Nutzen Sie PL/SQL bis zur Neige .....	952
21.1.8	Kontrollieren Sie den Speicherverbrauch .....	953
21.1.9	Glauben Sie nicht an Wunder .....	953
21.1.10	Salvatorische Klausel .....	954
<b>21.2</b>	<b>Optimierungsmöglichkeiten von PL/SQL .....</b>	<b>955</b>
21.2.1	Automatisierte Codeoptimierung .....	955
21.2.2	Subprogram-Inlining .....	956
21.2.3	Native Kompilierung .....	957
21.2.4	Caching .....	960
21.2.5	Feingranulare Abhängigkeitsverwaltung .....	969
<b>21.3</b>	<b>Compiler-Warnungen .....</b>	<b>969</b>
<b>21.4</b>	<b>PL/Scope .....</b>	<b>973</b>
21.4.1	Welche Information bietet PL/Scope? .....	974
21.4.2	Die View »USER_IDENTIFIERS« .....	976
21.4.3	Die View »USER_STATEMENTS« .....	978
21.4.4	Administration von PL/Scope .....	980
<b>21.5</b>	<b>PL/SQL Hierarchical Profiler .....</b>	<b>980</b>
21.5.1	Der hierarchische Profiler im SQL Developer .....	981
21.5.2	Voraussetzungen für den Einsatz des hierarchischen Profilers .....	983
21.5.3	Das Package »DBMS_HPROF« .....	984
21.5.4	Die Analyse .....	986
21.5.5	Ein etwas realitätsnäheres Beispiel .....	988
21.5.6	Umgehung der Limitierungen .....	992
21.5.7	Optionen der Funktion »dbms_hprof.analyze« .....	998
<b>21.6</b>	<b>Den Speicherverbrauch von PL/SQL überwachen .....</b>	<b>999</b>
21.6.1	Die Speicherverwaltung von PL/SQL .....	1000
21.6.2	Überwachung des Arbeitsspeichers .....	1001

## 22 Workshop: PL/SQL Instrumentation Toolkit (PIT) 1003

---

<b>22.1</b>	<b>Überblick: die Idee und die Architektur .....</b>	<b>1003</b>
22.1.1	Funktionsumfang .....	1005
22.1.2	Anwendungsbeispiel .....	1006
22.1.3	Die beteiligten Komponenten .....	1009
22.1.4	Idee und Arbeitsweise .....	1010



- 22.2 Beschreibung der einzelnen Komponenten ..... 1012**
  - 22.2.1 Meldung ..... 1013
  - 22.2.2 Call-Stack ..... 1017
  - 22.2.3 Kontext ..... 1021
  - 22.2.4 Adapter ..... 1024
  - 22.2.5 Das Package »MSG« ..... 1026
  - 22.2.6 Ausgabemodul ..... 1027
  - 22.2.7 Die PIT-API ..... 1029
  - 22.2.8 Internationalisierung ..... 1031
  - 22.2.9 Die zentrale Komponente »PIT\_INTERNAL« ..... 1033
  - 22.2.10 Ein konkretes Ausgabemodul ..... 1039
- 22.3 Implementierung des PIT-Administrations-Packages ..... 1047**
  - 22.3.1 Funktionsüberblick und Implementierungsstrategie ..... 1047
  - 22.3.2 Implementierungsdetails ..... 1050
- 22.4 Weitere Ausgabemodule ..... 1055**
  - 22.4.1 Ausgabe in eigene Fehlerdateien ..... 1055
  - 22.4.2 Ausgabe in APEX ..... 1057
  - 22.4.3 Ausgabe in Alert-Log- oder Trace-Dateien ..... 1060
  - 22.4.4 Ausgabe in Logging-Tabellen ..... 1062
  - 22.4.5 Meldung als E-Mail versenden ..... 1062
  - 22.4.6 Meldungen in JMS integrieren ..... 1064
- Index ..... 1071**