

Auf einen Blick

1	Einführung	19
2	Verwendete Werkzeuge und Ressourcen	33

TEIL I Grundlagen

3	Aufbau der Datenbank aus Sicht eines Programmierers	59
4	Datenbankobjekte und SQL	117
5	Datensicherheit, -konsistenz und Transaktion	161
6	Programmierung der Datenbank	215

TEIL II Die Sprache PL/SQL

7	Die Blockstruktur und Syntax von PL/SQL	241
8	Kontrollstrukturen	283
9	Datentypen in PL/SQL	305
10	Dynamisches SQL	373
11	Events in der Datenbank: Programmierung von Triggern	391
12	Packages	455
13	Exception	521

TEIL III PL/SQL im Einsatz

14	Erweiterung von SQL	551
15	Arbeiten mit LOBs (Large Objects)	599
16	Arbeiten mit XML	635
17	Objektorientierung	711
18	Integration von Oracle in Applikationen	791
19	Code-Analyse und Performance-Tuning	855
20	Workshop: PL/SQL Instrumentation Toolkit (PIT)	903

Inhalt

1	Einführung	19
1.1	Für wen ist dieses Buch geschrieben?	19
1.2	Der Aufbau des Buches	22
1.2.1	Teil 1: Grundlagen	23
1.2.2	Teil 2: Die Sprache PL/SQL	24
1.2.3	Teil 3: PL/SQL im Einsatz	26
1.3	Vorwort zur zweiten Auflage	29
1.4	Danksagung	31
2	Verwendete Werkzeuge und Ressourcen	33
2.1	Oracles Online-Dokumentation	33
2.1.1	Wo finde ich die benötigten Informationen?	34
2.1.2	PL/SQL-Grundlagen	37
2.1.3	Oracle-Packages	37
2.1.4	Weiterführende Literatur	38
2.2	Aufsetzen einer Beispieldatenbank	39
2.3	SQL*Plus	42
2.4	SQL Developer	43
2.5	»explain plan«	45
2.6	Autotrace	46
2.7	RunStats	49
2.8	Trace und TKProf	50
2.9	Debugger	54
2.10	Weitere Werkzeuge	55
2.11	Die Beispielskripte	55

TEIL I Grundlagen

3 Aufbau der Datenbank aus Sicht eines Programmierers

59

3.1	Grundlegende Arbeitsweise der Datenbank	59
3.1.1	Anforderungen an ein Datenbank-Managementsystem	60
3.1.2	Die Begriffe Datenbank, Schema und Tablespace	64
3.1.3	Systemtabellen, Data Dictionary und Privilegien	65
3.1.4	Die Sicht der Anwendungsprogrammierer auf die Datenbank	67
3.2	Logischer Aufbau: Schema, Tablespace & Co.	69
3.2.1	Schema	70
3.2.2	Tablespace	74
3.2.3	Auswirkungen auf die Architektur einer Applikation	77
3.3	Die physikalische Datenbank	80
3.3.1	Datendateien	80
3.3.2	Redo-Log-Dateien	81
3.3.3	Kontrolldatei	82
3.3.4	Parameterdatei	83
3.3.5	Passwortdatei	84
3.3.6	FRA (Fast Recovery Area)	85
3.3.7	ASM	85
3.4	Instanz und Speicherstrukturen	86
3.4.1	Die Speicherbereiche der SGA	88
3.4.2	Shared Pool	90
3.4.3	Die Hintergrundprozesse	91
3.5	Start der Datenbank	96
3.6	Verbindungsaufbau zur Datenbank	97
3.6.1	Verbindungsarten und Treiber	99
3.6.2	DEDICATED-SERVER-Verbindung	106
3.6.3	Shared-Server-Verbindung	108
3.6.4	Database Resident Connection Pool	110
3.6.5	Und nun? Entscheidungshilfen für den Verbindungsaufbau	112

4 Datenbankobjekte und SQL 117

4.1 Tabellen	117
4.1.1 Heap Organized Table	117
4.1.2 Index Organized Table	119
4.1.3 Temporäre Tabellen	120
4.1.4 Partitionierte Tabellen	121
4.2 Index	123
4.2.1 Anmerkung zur Benutzung von Indizes	126
4.2.2 B*-Baum-Index	128
4.2.3 Reverse-Key-Index	130
4.2.4 Funktionsbasierter Index	130
4.3 Views und Materialized Views	132
4.3.1 Views	132
4.3.2 Materialized Views	134
4.4 PL/SQL-Programm	135
4.5 Sonstige Datenbankobjekte	136
4.5.1 Sequenzen	136
4.5.2 Synonym	138
4.5.3 Database Link	138
4.5.4 Große Datenmengen: »CLOB«, »NCLOB«, »BLOB« und »BFile«	139
4.5.5 Benutzerdefinierte Typen, XML	141
4.5.6 Weitere Datenbankobjekte	141
4.6 Exkurs: Zeichensatzkodierung	141
4.6.1 Zeichensatzkodierung im Überblick	142
4.6.2 Zeichensatzkodierung bei Oracle	143
4.7 Mächtigkeit von SQL	147
4.7.1 Analytische Funktionen	148
4.7.2 Hierarchische Abfragen	151
4.7.3 Error Logging	154
4.7.4 Fazit	158

5 Datensicherheit, -konsistenz und Transaktion 161

5.1 Lese- und Schreibkonsistenz	162
5.1.1 Lesekonsistenz	162
5.1.2 Schreibkonsistenz	166

- 5.2 **Transaktion** 166
 - 5.2.1 Transaktion zum Schutz der Lesekonsistenz 166
 - 5.2.2 Transaktion zur Definition eines Geschäftsvorfalles 168
 - 5.2.3 Zusammenfassung 169
- 5.3 **Datenkonsistenz und referenzielle Integrität** 171
 - 5.3.1 Datenintegrität 171
 - 5.3.2 Performance-Überlegungen zu Datenbank-Constraints 177
 - 5.3.3 Datenkonsistenz 180
 - 5.3.4 Zusammenfassung 184
- 5.4 **Explizites Sperren von Daten durch die Anwendung** 184
 - 5.4.1 Das Problem: Lost Updates 184
 - 5.4.2 Das optimistische Sperren 186
 - 5.4.3 Das pessimistische Sperren 189
 - 5.4.4 Das vorsichtig optimistische Sperren 189
 - 5.4.5 Und nun? Wann sollte welche Sperrstrategie verwendet werden? ... 190
- 5.5 **Verarbeitung einer SQL-Anweisung** 191
 - 5.5.1 Parsen und Optimierung 192
 - 5.5.2 Datenlieferung über Cursor 197
- 5.6 **Die Sperrmechanismen von Oracle** 198
 - 5.6.1 Locks 198
 - 5.6.2 Latches 199
- 5.7 **Datensicherheit** 199
- 5.8 **Workshop: Einfluss der Programmierung** 202
 - 5.8.1 Das Ziel unserer Programmierung 202
 - 5.8.2 Implementierung des Tests 204

6 Programmierung der Datenbank 215

- 6.1 **Erweiterung der Datenbankfunktionalität** 215
- 6.2 **Programmierung der Datenkonsistenz** 217
 - 6.2.1 Datenbanktrigger 217
 - 6.2.2 Datenzugriff über PL/SQL 221
 - 6.2.3 Datenkonsistenz jenseits referenzieller Integrität 223
- 6.3 **Programmierung der Datensicherheit** 224
- 6.4 **Anwendungsprogrammierung mit PL/SQL** 228
 - 6.4.1 PL/SQL auf der Client-Seite 228
 - 6.4.2 Webanwendungen mit PL/SQL entwickeln 230

6.5	Unterstützung der Administration durch PL/SQL	234
6.5.1	Einsatz von PL/SQL in Skripten	235
6.5.2	Verwaltung wiederkehrender Aufgaben mit Scheduler und Jobs	236
6.5.3	Datenbanktrigger im Umfeld der Datensicherung und des Auditing	237

TEIL II Die Sprache PL/SQL

7 Die Blockstruktur und Syntax von PL/SQL 241

7.1	Das Grundgerüst: der PL/SQL-Block	242
7.1.1	Deklaration von Variablen	246
7.1.2	Schachtelung von Blöcken zur Fehlerbehandlung	248
7.1.3	Gültigkeitsbereich von Variablen	249
7.2	Prozeduren	249
7.2.1	Prozeduren mit Parametern	253
7.2.2	Formen der Parameterzuweisung	259
7.2.3	Optionale Parameter	260
7.3	Funktionen	264
7.4	Datenbanktrigger	267
7.5	Packages	270
7.5.1	Package-Spezifikation	270
7.5.2	Package-Körper	272
7.5.3	Aufruf von Prozeduren und Methoden in Packages	274
7.6	Ausführungsrechte von PL/SQL-Blöcken	275
7.7	Compiler-Anweisungen (Pragma)	278
7.7.1	Die autonome Transaktion	278
7.7.2	Initialisierung eigener Fehler	279
7.8	Best Practices	280

8 Kontrollstrukturen 283

8.1	Auswertende Anweisung 1 (»if then else«-Anweisung)	283
8.2	Auswertende Anweisung 2 (CASE-Anweisung)	285
8.2.1	Einfache »case«-Anweisung	285

8.2.2	Die auswertende »case«-Anweisung	286
8.3	Einfache Schleifen	288
8.3.1	Basisschleife (Schleife)	288
8.3.2	Abweisende Schleife 1 (FOR-Schleife)	290
8.3.3	Abweisende Schleife 2 (WHILE-Schleife)	292
8.3.4	Best Practices	293
8.4	Konditionale Kompilierung	295
8.4.1	Die AuswahlDirektive (selection directive)	297
8.4.2	Die Abfragedirektive (inquiry directive)	297
8.4.3	Die Error-Direktive (error directive)	300
8.5	Aus der Mottenkiste: Konzepte, die Sie nicht verwenden sollten	301
8.5.1	Label	301
8.5.2	»continue«- und »goto«-Anweisung	303

9 Datentypen in PL/SQL 305

9.1	PL/SQL-Datentypen	305
9.1.1	SQL-Datentypen	305
9.1.2	Basistypen und Subtypen in PL/SQL	308
9.1.3	SQL-Datentypen mit abweichender Definition in PL/SQL	310
9.1.4	SQL-Datentypen, die in PL/SQL nicht existieren	311
9.1.5	PL/SQL-Datentypen, die in SQL nicht existieren	311
9.1.6	Benutzerdefinierte Datentypen	312
9.1.7	Ableitung von Variablentypen aus dem Data Dictionary	312
9.2	Kollektionen in PL/SQL	315
9.2.1	Record	315
9.2.2	Assoziative Tabellen	325
9.2.3	VARRAY oder NESTED _TABLE als Alternative zu einer assoziativen Tabelle	327
9.2.4	Massenverarbeitung mit assoziativen Tabellen	328
9.3	Cursor	335
9.3.1	Lebenszyklus eines Cursors	336
9.3.2	Cursor-Attribute	338
9.3.3	Parametrierte Cursor	342
9.3.4	Mengenverarbeitung mit »bulk collect«	343
9.3.5	Kurzform: die »cursor for«-Schleife	345
9.3.6	Implizite versus explizite Cursor	347
9.3.7	Cursor-Variablen (REF-Cursor)	352

9.3.8	Cursor-Ausdrücke	356
9.3.9	Gemeinsamer Zugriff auf Daten über verteilte Cursor	362
9.4	Pipelined Function	363
9.4.1	Was ist eine Pipelined Function?	364
9.4.2	Workshop: Pipelined Function	365
9.4.3	Workshop: Verwendung von Pipelined Functions	369
9.4.4	Workshop: Erstellung einer Pipelined Function	370

10 Dynamisches SQL 373

10.1	Dynamisches SQL	373
10.1.1	Verwendung von Bindevariablen	374
10.2	Dynamisches SQL mit Cursor-Variablen	377
10.3	Workshop: Erstellung einer Prozedur als Schnittstelle zu einem externen Programm	377
10.4	DBMS_SQL-Package	382
10.5	Sicherheit bei dynamischem SQL	385
10.5.1	SQL-Injection über Suchparameter	385
10.5.2	SQL-Injection über Formatangaben	386
10.5.3	SQL-Injection über das Einschmuggeln zusätzlicher Anweisungen ...	387
10.5.4	Vermeidung von SQL-Injection 1: Bindevariablen	388
10.5.5	Vermeidung von SQL-Injection 2: »dbms_assert«	388

11 Events in der Datenbank: Programmierung von Triggern 391

11.1	DML-Trigger	391
11.1.1	Anweisungs- versus Zeilentrigger	392
11.1.2	Der Triggerkörper	397
11.1.3	Wann wird ein Trigger ausgelöst?	397
11.1.4	Das Mutating-Table-Problem	401
11.1.5	Compound Trigger	404
11.1.6	Workshop: Lösung des Mutating-Table-Problems mit einem Compound Trigger	405
11.1.7	Cross Edition Trigger	411
11.1.8	Benennungskonvention von Triggern	411

11.2 Einsatzbereiche von DML-Triggern	412
11.2.1 Erweiterung der Datenkonsistenzprüfung über Constraints hinaus	412
11.2.2 Workshop: Statusänderungen in einer bestimmten Reihenfolge durchführen	414
11.2.3 Implementierung einfacher Geschäftsregeln	417
11.2.4 Historisierung, Logging und Auditing von Daten	419
11.2.5 Workshop: Generisches Logging	420
11.3 »instead of«-Trigger	428
11.3.1 Deklaration eines »instead of«-Triggers	428
11.3.2 Workshop: Historisierung von Daten mit einem »instead of«-Trigger	430
11.4 Wann Sie DML-Trigger nicht verwenden sollten	437
11.5 Auditing mithilfe von Triggern	439
11.6 Datenbanktrigger	440
11.6.1 Ereignisattribute	442
11.6.2 Datenbankereignisse	445
11.6.3 Benutzerbezogene Ereignisse	446
11.6.4 DDL-Ereignisse	450
11.6.5 Systemereignisse	452
11.7 Zusammenfassung	452
 12 Packages	 455
12.1 Trennung von öffentlicher und privater Logik	455
12.1.1 Deklaration	457
12.1.2 Implementierung	458
12.1.3 Zusammenfassung	466
12.2 Überladung in Packages	467
12.2.1 Deklaration	468
12.2.2 Implementierung	470
12.2.3 Zusammenfassung	474
12.3 Packages und die Dependency Chain	475
12.4 Verschlüsselung von Package-Code	481
12.4.1 Das »wrap«-Utility	482
12.4.2 Verwendung des Packages »dbms_ddl«	484

12.5 Oracle-Packages	487
12.5.1 Das Package »standard«	489
12.5.2 Wichtige Oracle-Packages	491
12.6 Workshop: Verwaltung von Anwendungsparametern	500
12.6.1 Das Problem und die Lösungsidee	500
12.6.2 Vorüberlegungen zur Parametertabelle	501
12.6.3 Die Parametertabelle	504
12.6.4 Einrichtung der Parametertabelle und der Zugriffsrechte	506
12.6.5 Das Parameter-Package	510
12.6.6 Das Package im Einsatz	516

13 Exception 521

13.1 Oracle-Fehler	521
13.1.1 Benannte Fehler	525
13.1.2 »SQLERRM«- und »SQLCODE«-Funktionen und der Fehler-Stack	527
13.1.3 Nicht benannte Fehler benennen	530
13.2 Applikationsfehler erstellen und bearbeiten	531
13.2.1 Fehler direkt mit »RAISE_APPLICATION_ERROR« erzeugen	531
13.2.2 Fehler aus einem Fehler-Package erstellen lassen	532
13.2.3 Zentralisierung der Fehlermeldungen über »LMSGEN«	534
13.2.4 Workshop: Wrapper-Package um »utl_lms«	537
13.2.5 Zusammenfassung: Fehlermeldungen mit »utl_lms«	541
13.3 Workshop: Zentralisierter Fehler-Handler mit einem Trigger	541
13.4 Zusammenfassung	547

TEIL III PL/SQL im Einsatz

14 Erweiterung von SQL 551

14.1 Wann SQL erweitert werden sollte	551
14.1.1 Bleiben Sie auf dem aktuellen Wissensstand	551
14.1.2 Voraussetzungen für die Erweiterung von SQL	555
14.2 SQL mit eigenen Funktionen erweitern	557
14.2.1 Anforderungen an den PL/SQL-Block	558
14.2.2 Nebenwirkungsfreiheit (Purity)	558

14.2.3	Optimizer Hints	559
14.2.4	Das Pragma »restrict_references«	560
14.2.5	Workshop: Deterministische Funktion	560
14.3	Workshop: Berechnung der Fakultät	567
14.3.1	Einschränkung der Fakultätsfunktion auf definierte Werte	568
14.3.2	Zielvorgabe	569
14.3.3	Und was ist mit Rekursion?	575
14.4	Gruppenfunktionen selbst erstellen	576
14.4.1	Arbeitsweise von Gruppenfunktionen	577
14.4.2	Workshop: Erstellung einer Gruppenfunktion	581
14.4.3	Test der Gruppenfunktion	585
14.4.4	Zusammenfassung	587
14.5	Workshop: Code-Generator für Gruppenfunktionen	587
15	Arbeiten mit LOBs (Large Objects)	599
<hr/>		
15.1	Technische Struktur	600
15.1.1	Einsatz von LOB-Datentypen in der Datenbank	600
15.1.2	LOB als PL/SQL-Variable	605
15.1.3	LOB als Methodenparameter	612
15.1.4	SecureFiles	613
15.2	Die Datentypen »CLOB«, »NCLOB«, »BLOB« und »BFILE«	616
15.2.1	»CLOB« und »NCLOB«	616
15.2.2	Der binäre Datentyp »BLOB«	617
15.2.3	»BFile«	617
15.3	Das Package »DBMS_LOB«	619
15.3.1	Schreibzugriff auf temporäre oder persistente LOBs	620
15.3.2	Verwaltung temporärer und persistenter LOBs	622
15.3.3	API für »BFile«-LOB	624
15.3.4	Zugriff auf LOBs durch die Anwendung	625
15.4	Workshop: Hilfsfunktionen zum Arbeiten mit LOBs	625
15.4.1	Hilfsfunktion zum Laden von »CLOB« und »BLOB« aus dem Dateisystem in die Datenbank	625
15.4.2	Oder aber	634

16 Arbeiten mit XML 635

16.1 Der Datentyp »XMLType«	635
16.1.1 Verwendung von »XMLType« als Tabellen- oder Spaltentyp	636
16.1.2 »XMLType«-Member Functions	639
16.1.3 Umformung von XML mittels XSLT	640
16.2 Die Speicherung von XML-Daten in der Datenbank	645
16.3 XML aus relationalen Daten erzeugen	648
16.3.1 Der SQL/XML-Standard	648
16.3.2 Das Package »dbms_xmlgen« bzw. »sys_xmlgen«	653
16.4 Relationale Daten aus XML extrahieren	663
16.4.1 Extraktion relationaler Daten mit »XMLTable«	663
16.4.2 Extraktion relationaler Daten mittels Objektorientierung	667
16.4.3 Extraktion relationaler Daten mittels Java-Programmierung	669
16.5 XML mit PL/SQL verarbeiten	670
16.5.1 Die Programmierung mittels DOM-Baum	670
16.5.2 Die XML-Packages	672
16.6 Die XML-Datenbank	683
16.6.1 Einführung in die XML-Datenbank	684
16.6.2 Speicherung und Veröffentlichung binärer Dokumente und XML-Dokumente	686
16.6.3 Dokumente über XDB verwalten	690
16.6.4 Zugriffsschutz und Sicherheit der XDB	700
16.6.5 Versionierung von Ressourcen	706

17 Objektorientierung 711

17.1 Einführung in die Objektorientierung	713
17.1.1 Alles ist ein Objekt	713
17.1.2 Das zweite Reizwort: Vererbung!	715
17.1.3 Abstrakte und finale Klassen	717
17.1.4 Statische Methoden	718
17.1.5 Objektidentität versus »Statement of Truth«	718
17.1.6 Klassen haben komplexe Strukturen	720
17.1.7 Auswirkungen auf die Datenbankprogrammierung	722
17.2 Objektorientierte Datentypen	724
17.2.1 »OBJECT«	725

17.2.2	»VARRAY«	726
17.2.3	»NESTED TABLE«	729
17.2.4	Vergleiche von Kollektionen	730
17.2.5	Methoden von Kollektionstypen	731
17.2.6	Workshop: Liste von Werten übergeben	732
17.3	Objektorientierte Datenmodelle	735
17.4	Workshop: Der Datentyp »MoneyType«	739
17.4.1	Vorüberlegungen	739
17.4.2	Implementierung des Typs »MoneyType«	740
17.4.3	Der Typkörper	743
17.4.4	Implementierung des Packages »moneytype_pkg«	745
17.4.5	Der Package-Körper	746
17.4.6	Die Rechtesituation ab Version 11g	755
17.4.7	Erweiterung durch Vererbung	758
17.5	Objektorientierte Anwendungsentwicklung und relationale Datenbanken	760
17.5.1	Das Problem des Impedance Mismatch	761
17.5.2	Lösungsansatz 1: Die Vision der generischen Datenbank	771
17.5.3	Lösungsansatz 2: Objektrelationale Mapping-Werkzeuge	778
17.5.4	Lösungsansatz 3: Die Datenbank als Daten-Framework	782
18	Integration von Oracle in Applikationen	791
18.1	Sperrung von Daten bei der Datenänderung	792
18.1.1	Transaktionsschutz innerhalb der Datenbank	793
18.1.2	Erweiterter Fokus: Datensicherung im Umfeld von Anwendungen ...	793
18.1.3	Pessimistisches Locking	794
18.1.4	Optimistisches Sperren	800
18.1.5	Database-Change-Notification-basiertes Locking	808
18.2	Speicherung von Session-Informationen	810
18.2.1	Grundlagen eines Kontextes	811
18.2.2	Session-Kontext	812
18.2.3	Globally Accessed Context	814
18.2.4	Workshop: Package zur Verwaltung von Kontexten	817
18.3	Zugriff auf Daten über PL/SQL-Packages	829
18.3.1	Kapselung von DML-Operationen in Packages	830
18.3.2	Vermeidung von Triggern durch Packages	831
18.3.3	Integration datenbezogener Geschäftsregeln	832

18.4	Workshop: Keimzelle einer sicheren Datenbankanwendung	834
18.4.1	Das Projekt	834
18.4.2	Übersicht über die Architektur	835
18.4.3	Die »logon«-Prozedur	838
18.4.4	Aufsetzen der Schemata	839
18.4.5	Die Packages	844
18.4.6	Test der Architektur	850
18.4.7	Zusammenfassung und Ausblick	852

19	Code-Analyse und Performance-Tuning	855
-----------	--	------------

19.1	Optimierungsmöglichkeiten von PL/SQL	855
19.1.1	Automatisierte Code-Optimierung	856
19.1.2	Subprogram-Inlining	857
19.1.3	Native Kompilierung	858
19.1.4	Caching	860
19.1.5	Feingranulare Abhängigkeitsverwaltung	867
19.2	Compiler-Warnungen	867
19.3	PL/Scope	871
19.3.1	Welche Information bietet PL/Scope?	872
19.3.2	Die View »USER_IDENTIFIERS«	874
19.3.3	Administration von PL/Scope	876
19.4	PL/SQL Hierarchical Profiler	877
19.4.1	Der hierarchische Profiler im SQL Developer	878
19.4.2	Voraussetzungen für den Einsatz des hierarchischen Profilers	880
19.4.3	Das Package »DBMS_HPROF«	881
19.4.4	Die Analyse	883
19.4.5	Ein etwas realitätsnäheres Beispiel	885
19.4.6	Optionen der Funktion »dbms_hprof.analyze«	889
19.5	Den Speicherverbrauch von PL/SQL überwachen	890
19.5.1	Die Speicherverwaltung von PL/SQL	891
19.5.2	Überwachung des Arbeitsspeichers	892
19.6	Regeln zur Performance-Optimierung	893
19.6.1	Nutzen Sie SQL, falls dies möglich ist	894
19.6.2	Betrachten Sie die Datenbank als entfernte Ressource	896
19.6.3	Benutzen Sie Bindevariablen	896
19.6.4	Arbeiten Sie mengenorientiert	897
19.6.5	Arbeiten Sie sich in die Grundkonzepte der Datenbank ein	897

20 Workshop: PL/SQL Instrumentation Toolkit (PIT) 90318