

Inhalt

1	Einführung	21
1.1	Für wen ist dieses Buch geschrieben?	21
1.2	Der Aufbau des Buches	24
1.2.1	Teil 1: Grundlagen	25
1.2.2	Teil 2: Die Sprache PL/SQL	27
1.2.3	Teil 3: PL/SQL im Einsatz	29
1.3	Vorwort zur zweiten Auflage	32
1.4	Vorwort zur dritten Auflage	33
1.5	Danksagung	34
2	Verwendete Werkzeuge und Ressourcen	35
2.1	Oracles Online-Dokumentation	35
2.1.1	Wo finde ich die benötigten Informationen?	36
2.1.2	PL/SQL-Grundlagen	38
2.1.3	Oracle-Packages	39
2.1.4	Weiterführende Literatur	40
2.2	Aufsetzen einer Beispieldatenbank	41
2.3	SQL*Plus	43
2.4	SQLCL	45
2.5	SQL Developer	45
2.6	»explain plan«	46
2.7	Autotrace	48
2.8	RunStats	50
2.9	Trace und TKProf	51
2.10	Debugger	55
2.11	Weitere Werkzeuge	56
2.12	Die Beispielskripte	57

TEIL I Grundlagen

3 Aufbau der Datenbank aus Sicht eines Programmierers

61

3.1	Grundlegende Arbeitsweise der Datenbank	61
3.1.1	Anforderungen an ein Datenbank-Managementsystem	62
3.1.2	Lesekonsistenz	65
3.1.3	Die Begriffe Datenbank, Schema und Tablespace	65
3.1.4	Systemtabellen, Data Dictionary und Privilegien	67
3.1.5	Die Sicht der Anwendung auf die Datenbank	68
3.2	Logischer Aufbau: Schema, Tablespace & Co.	71
3.2.1	Schema	71
3.2.2	Tablespace	76
3.2.3	Auswirkungen auf die Architektur einer Applikation	79
3.3	Die physikalische Datenbank	82
3.3.1	Datendateien	82
3.3.2	Redo-Log-Dateien	84
3.3.3	Kontrolldatei	85
3.3.4	Parameterdatei	85
3.3.5	Passwortdatei	86
3.3.6	FRA (Fast Recovery Area)	87
3.3.7	ASM	88
3.4	Instanz und Speicherstrukturen	88
3.4.1	Die Speicherbereiche der SGA	90
3.4.2	Shared Pool	93
3.4.3	Die Hintergrundprozesse	94
3.5	Start der Datenbank	98
3.6	Verbindungsaufbau zur Datenbank	99
3.6.1	Verbindungsarten und Treiber	102
3.6.2	Dedicated-Server-Verbindung	109
3.6.3	Shared-Server-Verbindung	110
3.6.4	Database Resident Connection Pool	112
3.6.5	Und nun? Entscheidungshilfen für den Verbindungsaufbau	115

4 Datenbankobjekte und SQL 119

4.1 Tabellen	119
4.1.1 Heap Organized Table	119
4.1.2 Index Organized Table	121
4.1.3 Temporäre Tabellen	122
4.1.4 Partitionierte Tabellen	123
4.2 Index	125
4.2.1 Anmerkung zur Benutzung von Indizes	128
4.2.2 B*-Baum-Index	129
4.2.3 Reverse-Key-Index	131
4.2.4 Funktionsbasierter Index	132
4.3 Views und Materialized Views	134
4.3.1 Views	134
4.3.2 Materialized Views	135
4.4 PL/SQL-Programm	137
4.5 Sonstige Datenbankobjekte	138
4.5.1 Sequenzen	138
4.5.2 Synonym	139
4.5.3 Database Link	140
4.5.4 Große Datenmengen: CLOB, NCLOB, BLOB und BFile	141
4.5.5 Benutzerdefinierte Typen, XML, JSON	143
4.5.6 Weitere Datenbankobjekte	143
4.6 Exkurs: Zeichensatzcodierung	144
4.6.1 Zeichensatzcodierung im Überblick	144
4.6.2 Zeichensatzcodierung bei Oracle	146
4.7 Mächtigkeit von SQL	150
4.7.1 Analytische Funktionen	150
4.7.2 Hierarchische Abfragen	153
4.7.3 Error Logging	156
4.7.4 Fazit	161

5 Datensicherheit, -konsistenz und Transaktion 163

5.1 Lese- und Schreibkonsistenz	164
5.1.1 Lesekonsistenz	164
5.1.2 Schreibkonsistenz	167

5.2	Transaktion	168
5.2.1	Transaktion zum Schutz der Lesekonsistenz	168
5.2.2	Transaktion zur Definition eines Geschäftsvorfalles	170
5.2.3	Zusammenfassung	171
5.3	Datenkonsistenz und referenzielle Integrität	172
5.3.1	Datenintegrität	173
5.3.2	Performance-Überlegungen zu Datenbank-Constraints	178
5.3.3	Datenkonsistenz	182
5.3.4	Zusammenfassung	185
5.4	Explizites Sperren von Daten durch die Anwendung	186
5.4.1	Das Problem: Lost Updates	186
5.4.2	Das optimistische Sperren	187
5.4.3	Das pessimistische Sperren	190
5.4.4	Das vorsichtig optimistische Sperren	191
5.4.5	Und nun? Wann sollte welche Sperrstrategie verwendet werden?	191
5.5	Verarbeitung einer SQL-Anweisung	193
5.5.1	Parsen und Optimierung	193
5.5.2	Datenlieferung über Cursor	198
5.6	Die Sperrmechanismen von Oracle	199
5.6.1	Locks	199
5.6.2	Latches	200
5.7	Datensicherheit	200
5.8	Workshop: Einfluss der Programmierung	203
5.8.1	Das Ziel unserer Programmierung	204
5.8.2	Implementierung des Tests	205

6 Programmierung der Datenbank 217

6.1	Erweiterung der Datenbankfunktionalität	217
6.2	Programmierung der Datenkonsistenz	219
6.2.1	Datenbanktrigger	219
6.2.2	Datenzugriff über PL/SQL	223
6.2.3	Datenkonsistenz jenseits referenzieller Integrität	226
6.3	Programmierung der Datensicherheit	226

6.4	Anwendungsprogrammierung mit PL/SQL	230
6.4.1	PL/SQL auf der Client-Seite	230
6.4.2	Webanwendungen mit PL/SQL entwickeln	232
6.5	Unterstützung der Administration durch PL/SQL	236
6.5.1	Einsatz von PL/SQL in Skripten	237
6.5.2	Verwaltung wiederkehrender Aufgaben mit Scheduler und Jobs	238
6.5.3	Datenbanktrigger im Umfeld der Datensicherung und des Auditings	238

TEIL II Die Sprache PL/SQL

7 Die Blockstruktur und Syntax von PL/SQL 243

7.1	Das Grundgerüst: der PL/SQL-Block	244
7.1.1	Deklaration von Variablen	248
7.1.2	Schachtelung von Blöcken zur Fehlerbehandlung	249
7.1.3	Gültigkeitsbereich von Variablen	250
7.2	Prozeduren	251
7.2.1	Prozeduren mit Parametern	255
7.2.2	Formen der Parameterzuweisung	261
7.2.3	Optionale Parameter	262
7.2.4	Beliebig viele Parameter an eine Methode übergeben	266
7.3	Funktionen	267
7.4	Datenbanktrigger	271
7.5	Packages	272
7.5.1	Package-Spezifikation	273
7.5.2	Package-Körper	274
7.5.3	Aufruf von Prozeduren und Methoden in Packages	277
7.6	Ausführungsrechte von PL/SQL-Blöcken	277
7.7	Compiler-Anweisungen (Pragma)	280
7.7.1	Die autonome Transaktion	280
7.7.2	Initialisierung eigener Fehler	281
7.8	Best Practices	282

8 Kontrollstrukturen 285

8.1	Auswertende Anweisung 1 (»if then else«-Anweisung)	285
8.2	Auswertende Anweisung 2 (»case«-Anweisung)	287
8.2.1	Einfache »case«-Anweisung	287
8.2.2	Aufruf der »case«-Anweisung als SQL-Ausdruck	288
8.2.3	Die auswertende »case«-Anweisung	289
8.3	Einfache Schleifen	291
8.3.1	Basisschleife (Schleife)	291
8.3.2	Abweisende Schleife 1 (»for«-Schleife)	293
8.3.3	Abweisende Schleife 2 (»while«-Schleife)	295
8.3.4	Best Practices	295
8.4	Konditionale Kompilierung	298
8.4.1	Die Auswahldirektive (Selection Directive)	299
8.4.2	Die Abfragedirektive (Inquiry Directive)	300
8.4.3	Die Error-Direktive (Error Directive)	303
8.5	Aus der Mottenkiste: Konzepte, die Sie nicht verwenden sollten	303
8.5.1	Label	304
8.5.2	»continue«- und »goto«-Anweisung	306

9 Datentypen in PL/SQL 309

9.1	PL/SQL-Datentypen	309
9.1.1	SQL-Datentypen	309
9.1.2	Basistypen und Subtypen in PL/SQL	312
9.1.3	SQL-Datentypen mit abweichender Definition in PL/SQL	314
9.1.4	SQL-Datentypen, die in PL/SQL nicht existieren	316
9.1.5	PL/SQL-Datentypen, die in SQL nicht existieren	316
9.1.6	Benutzerdefinierte Datentypen	316
9.1.7	Ableitung von Variablentypen aus dem Data Dictionary	317
9.2	Kollektionen in PL/SQL	319
9.2.1	Record	319
9.2.2	Assoziative Tabellen	330

9.2.3	Massenverarbeitung mit assoziativen Tabellen	332
9.2.4	VARRAY oder NESTED_TABLE als Alternative zu einer assoziativen Tabelle	339
9.3	Cursor	340
9.3.1	Lebenszyklus eines Cursors	341
9.3.2	Cursor-Attribute	343
9.3.3	Parametrisierte Cursor	347
9.3.4	Mengenverarbeitung mit »bulk collect«	348
9.3.5	Kurzform: die »cursor for«-Schleife	350
9.3.6	Implizite versus explizite Cursor	352
9.3.7	Cursor-Variablen (REF-Cursor)	357
9.3.8	Cursor-Ausdrücke	362
9.3.9	Gemeinsamer Zugriff auf Daten über verteilte Cursor	366
9.4	Pipelined Function	368
9.4.1	Was ist eine Pipelined Function?	368
9.4.2	Workshop: Pipelined Function	370
9.4.3	Verwendung von Pipelined Functions	373
9.4.4	Workshop: Erstellung einer Pipelined Function	374

10 Dynamisches SQL 379

10.1	Dynamisches SQL mittels »execute immediate«	379
10.1.1	Verwendung von Bindevariablen	380
10.2	Dynamisches SQL mit Cursor-Variablen	383
10.3	Workshop: Erstellung einer Prozedur als Schnittstelle zu einem externen Programm	384
10.4	DBMS_SQL-Package	389
10.4.1	Workshop: Code-Generator	392
10.5	Sicherheit bei dynamischem SQL	404
10.5.1	SQL-Injection über Suchparameter	405
10.5.2	SQL-Injection über Formatangaben	406
10.5.3	SQL-Injection über das Einschmuggeln zusätzlicher Anweisungen ...	406
10.5.4	Vermeidung von SQL-Injection 1: Bindevariablen	407
10.5.5	Vermeidung von SQL-Injection 2: »dbms_assert«	408

11	Events in der Datenbank: Programmierung von Triggern	411
11.1	DML-Trigger	411
11.1.1	Anweisungs- versus Zeilentrigger	412
11.1.2	Der Triggerkörper	417
11.1.3	Wann wird ein Trigger ausgelöst?	418
11.1.4	Das Mutating-Table-Problem	421
11.1.5	Compound Trigger	425
11.1.6	Workshop: Lösung des Mutating-Table-Problems mit einem Compound Trigger	426
11.1.7	Cross Edition Trigger	432
11.1.8	Benennungskonvention von Triggern	432
11.2	»instead of«-Trigger	432
11.3	Einsatzbereiche von DML-Triggern	435
11.3.1	Erweiterung der Datenkonsistenzprüfung über Constraints hinaus	435
11.3.2	Workshop: Statusänderungen in einer bestimmten Reihenfolge durchführen	437
11.3.3	Implementierung einfacher Geschäftsregeln	440
11.3.4	Historisierung, Logging und Auditing von Daten	442
11.3.5	Workshop: Generisches Logging	443
11.3.6	Workshop: Historisierung von Daten mit einem »instead of«-Trigger	451
11.4	Wann Sie DML-Trigger nicht verwenden sollten	458
11.4.1	Auditing mithilfe von Triggern	459
11.5	Datenbanktrigger	461
11.5.1	Ereignisattribute	463
11.5.2	Datenbankereignisse	466
11.5.3	Benutzerbezogene Ereignisse	467
11.5.4	DDL-Ereignisse	471
11.5.5	Systemereignisse	472
11.6	Zusammenfassung	473

12 Packages 477

12.1 Trennung von öffentlicher und privater Logik	477
12.1.1 Deklaration	478
12.1.2 Implementierung	479
12.1.3 Zusammenfassung	489
12.2 Überladung in Packages	490
12.2.1 Deklaration	491
12.2.2 Implementierung	493
12.2.3 Zusammenfassung	497
12.3 Ausführungsrechte von Packages	497
12.3.1 Rollen und Berechtigungskonzepte	499
12.3.2 Erweiterungen des Invokers-Rights-Berechtigungskonzepts	500
12.3.3 Steuerung des Zugriffs auf ein Package	501
12.4 Packages und die Dependency Chain	502
12.5 Verschlüsselung von Package-Code	509
12.5.1 Das »wrap«-Utility	509
12.5.2 Verwendung des Packages »dbms_ddl«	512
12.6 Oracle-Packages	514
12.6.1 Das Package »standard«	517
12.6.2 Wichtige Oracle-Packages	518
12.7 Workshop: Verwaltung von Anwendungsparametern	527
12.7.1 Das Problem und die Lösungs idee	527
12.7.2 Vorüberlegungen zur Parametertabelle	528
12.7.3 Die Parametertabelle	530
12.7.4 Einrichtung der Parametertabelle und der Zugriffsrechte	533
12.7.5 Das Parameter-Package	538
12.7.6 Das Package im Einsatz	544

13 Erweiterung von SQL 549

13.1 Wann SQL erweitert werden sollte	549
13.1.1 Bleiben Sie auf dem aktuellen Wissensstand	549
13.1.2 Voraussetzungen für die Erweiterung von SQL	554
13.2 SQL mit eigenen Funktionen erweitern	555
13.2.1 Anforderungen an den PL/SQL-Block	556
13.2.2 Nebenwirkungsfreiheit (Purity)	557

13.2.3	Optimizer Hints und Klauseln	557
13.2.4	Das Pragma »restrict_references«	558
13.2.5	Workshop: Deterministische Funktion	558
13.3	Workshop: Berechnung der Fakultät	565
13.3.1	Einschränkung der Fakultätsfunktion auf definierte Werte	566
13.3.2	Zielvorgabe	568
13.3.3	Und was ist mit Rekursion?	573
13.4	Gruppenfunktionen selbst erstellen	575
13.4.1	Arbeitsweise von Gruppenfunktionen	575
13.4.2	Workshop: Erstellung einer Gruppenfunktion	579
13.4.3	Test der Gruppenfunktion	583
13.4.4	Zusammenfassung	585
13.5	Workshop: Code-Generator für Gruppenfunktionen	585

14 Exception 603

14.1	Oracle-Fehler	603
14.1.1	Benannte Fehler	607
14.1.2	»sqlerrm«- und »sqlcode«-Funktionen und der Fehler-Stack	609
14.1.3	Nicht benannte Fehler benennen	615
14.2	Applikationsfehler erstellen und bearbeiten	616
14.2.1	Fehler direkt mit »raise_application_error« erzeugen	617
14.2.2	Fehler aus einem Fehler-Package erstellen lassen	618
14.2.3	Zentralisierung der Fehlermeldungen über »lmsgen«	620
14.2.4	Workshop: Wrapper-Package um »utl_lms«	623
14.2.5	Zusammenfassung: Fehlermeldungen mit »utl_lms«	626
14.3	Workshop: Zentralisierter Fehler-Handler mit einem Trigger	627
14.4	Zusammenfassung	632

TEIL III PL/SQL im Einsatz

15 Arbeiten mit LOBs (Large Objects) 635

15.1	Technische Struktur	636
15.1.1	Einsatz von LOB-Datentypen in der Datenbank	636
15.1.2	LOB als PL/SQL-Variable	641

15.1.3	LOB als Methodenparameter	648
15.1.4	SecureFiles	649
15.2	Die Datentypen »CLOB«, »NCLOB«, »BLOB« und »BFILE«	651
15.2.1	CLOB und NCLOB	652
15.2.2	Der binäre Datentyp »BLOB«	653
15.2.3	BFile	653
15.3	Das Package »DBMS_LOB«	655
15.3.1	Schreibzugriff auf temporäre oder persistente LOBs	656
15.3.2	Verwaltung temporärer und persistenter LOBs	658
15.3.3	API für BFile-LOB	659
15.3.4	Zugriff auf LOBs durch die Anwendung	661
15.4	Workshop: Hilfsfunktionen zum Arbeiten mit LOBs	661
15.4.1	Hilfsfunktion zum Laden von CLOB und BLOB aus dem Dateisystem in die Datenbank	661
15.4.2	Oder aber	670
16	Arbeiten mit XML und JSON	671
16.1	Der Datentyp »XMLType«	672
16.1.1	Verwendung von XMLType als Tabellen- oder Spaltentyp	672
16.1.2	»XMLType«-Member Functions	675
16.1.3	Umformung von XML mittels XSLT	677
16.2	Die Speicherung von XML-Daten in der Datenbank	680
16.3	XML aus relationalen Daten erzeugen	683
16.3.1	Der SQL/XML-Standard	684
16.3.2	Das Package »dbms_xmlgen«	689
16.4	Relationale Daten aus XML extrahieren	698
16.4.1	Extraktion relationaler Daten mit »XMLTable«	698
16.4.2	Extraktion relationaler Daten mittels Objektorientierung	702
16.4.3	Extraktion relationaler Daten mittels externer Programmierung	703
16.5	XML mit PL/SQL verarbeiten	704
16.5.1	Die Programmierung mittels DOM-Baum	705
16.5.2	Die XML-Packages	706
16.6	Die XML-Datenbank	716
16.6.1	Einführung in die XML-Datenbank	717
16.6.2	Speicherung und Veröffentlichung binärer Dokumente und XML-Dokumente	720

16.6.3	Dokumente über XDB verwalten	723
16.6.4	Zugriffsschutz und Sicherheit von XDB	732
16.6.5	Versionierung von Ressourcen	738
16.7	JSON	741
16.7.1	Überblick: Was ist JSON?	741
16.7.2	Der Datentyp JSON	743
16.7.3	Abfragen gegen JSON-Instanzen	746
16.7.4	Programmierung von JSON mit PL/SQL	750

17 Objektorientierung 759

17.1	Einführung in die Objektorientierung	761
17.1.1	Alles ist ein Objekt	761
17.1.2	Das zweite Reizwort: Vererbung!	763
17.1.3	Abstrakte und finale Klassen	765
17.1.4	Statische Methoden	766
17.1.5	Objektidentität versus »Statement of Truth«	766
17.1.6	Klassen haben komplexe Strukturen	768
17.1.7	Auswirkungen auf die Datenbankprogrammierung	770
17.2	Objektorientierte Datentypen	772
17.2.1	»object«	773
17.2.2	»varray«	774
17.2.3	»nested table«	777
17.2.4	Vergleiche von Kollektionen	778
17.2.5	Methoden von Kollektionstypen	779
17.2.6	Workshop: Liste von Werten übergeben	780
17.3	Objektorientierte Datenmodelle	783
17.4	Workshop: Der Datentyp »MoneyType«	787
17.4.1	Vorüberlegungen	787
17.4.2	Implementierung des Typs »MoneyType«	788
17.4.3	Der Typkörper	790
17.4.4	Implementierung des Packages »moneytype_pkg«	793
17.4.5	Der Package-Körper	794
17.4.6	Die Rechtesituation ab Version 11g	803
17.4.7	Erweiterung durch Vererbung	806
17.5	Objektorientierte Anwendungsentwicklung und relationale Datenbanken	808
17.5.1	Das Problem des Impedance Mismatch	809

17.5.2	Lösungsansatz 1: Die Vision der generischen Datenbank	819
17.5.3	Lösungsansatz 2: Objektrelationale Mapping-Werkzeuge	826
17.5.4	Lösungsansatz 3: Das »Thick Database«-Paradigma	829

18 Integration von Oracle in Applikationen 839

18.1	Sperrung von Daten bei der Datenänderung	840
18.1.1	Transaktionsschutz innerhalb der Datenbank	840
18.1.2	Erweiterter Fokus: Datensicherung im Umfeld von Anwendungen	841
18.1.3	Pessimistisches Locking	842
18.1.4	Optimistisches Sperren	848
18.1.5	Database-Change-Notification-basiertes Locking	856
18.2	Speicherung von Session-Informationen	858
18.2.1	Grundlagen eines Kontextes	859
18.2.2	Session-Kontext	859
18.2.3	Globally Accessed Context	861
18.2.4	Workshop: Package zur Verwaltung von Kontexten	865
18.3	Zugriff auf Daten über PL/SQL-Packages	877
18.3.1	Kapselung von DML-Operationen in Packages	877
18.3.2	Vermeidung von Triggern durch Packages	878
18.3.3	Integration datenbezogener Geschäftsregeln	879
18.4	Workshop: Keimzelle einer sicheren Datenbankanwendung	880
18.4.1	Das Projekt	881
18.4.2	Übersicht über die Architektur	881
18.4.3	Die »logon«-Prozedur	884
18.4.4	Aufsetzen der Schemata	884
18.4.5	Die Packages	890
18.4.6	Test der Architektur	896
18.4.7	Zusammenfassung und Ausblick	898

19 Performance-Tuning und Code-Analyse 901

19.1	Regeln zur Performance-Optimierung	902
19.1.1	Nutzen Sie SQL, falls dies möglich ist	902
19.1.2	Betrachten Sie die Datenbank als entfernte Ressource	905
19.1.3	Benutzen Sie Bindevariablen	905

19.1.4	Arbeiten Sie mengenorientiert	906
19.1.5	Arbeiten Sie sich in die Grundkonzepte der Datenbank ein	906
19.1.6	Nutzen Sie PL/SQL bis zur Neige	907
19.1.7	Kontrollieren Sie den Speicherverbrauch	908
19.1.8	Glauben Sie nicht an Wunder	908
19.1.9	Salvatorische Klausel	909
19.2	Optimierungsmöglichkeiten von PL/SQL	910
19.2.1	Automatisierte Code-Optimierung	910
19.2.2	Subprogram-Inlining	911
19.2.3	Native Kompilierung	912
19.2.4	Caching	915
19.2.5	Feingranulare Abhängigkeitsverwaltung	923
19.3	Compiler-Warnungen	923
19.4	PL/Scope	927
19.4.1	Welche Information bietet PL/Scope?	928
19.4.2	Die View »USER_IDENTIFIERS«	930
19.4.3	Die View »USER_STATEMENTS«	932
19.4.4	Administration von PL/Scope	934
19.5	PL/SQL Hierarchical Profiler	934
19.5.1	Der hierarchische Profiler im SQL Developer	935
19.5.2	Voraussetzungen für den Einsatz des hierarchischen Profilers	937
19.5.3	Das Package »DBMS_HPROF«	938
19.5.4	Die Analyse	940
19.5.5	Ein etwas realitätsnäheres Beispiel	942
19.5.6	Umgehung der Limitierungen	946
19.5.7	Optionen der Funktion »dbms_hprof.analyze«	952
19.6	Den Speicherverbrauch von PL/SQL überwachen	953
19.6.1	Die Speicherverwaltung von PL/SQL	954
19.6.2	Überwachung des Arbeitsspeichers	955
20	Workshop: PL/SQL Instrumentation Toolkit (PIT)	957
20.1	Überblick: die Idee und die Architektur	957
20.1.1	Funktionsumfang	959
20.1.2	Anwendungsbeispiel	960
20.1.3	Die beteiligten Komponenten	962
20.1.4	Idee und Arbeitsweise	964

20.2 Beschreibung der einzelnen Komponenten	966
20.2.1 Meldung	966
20.2.2 Call Stack	972
20.2.3 Kontext	976
20.2.4 Adapter	979
20.2.5 Das Package »MSG«	981
20.2.6 Ausgabemodul	982
20.2.7 Die PIT-API	984
20.2.8 Die zentrale Komponente »PIT_PKG«	986
20.2.9 Ein konkretes Ausgabemodul	995
20.2.10 Test des PIT	1004
20.3 Implementierung des PIT-Administrationspackages	1006
20.3.1 Funktionsüberblick und Implementierungsstrategie	1006
20.3.2 Spezifikation	1008
20.3.3 Package-Körper	1010
20.4 Weitere Ausgabemodule	1026
20.4.1 Ausgabe in eigene Fehlerdateien	1026
20.4.2 Ausgabe in APEX	1028
20.4.3 Ausgabe in Alert-Log- oder Trace-Dateien	1031
20.4.4 Ausgabe in Logging-Tabellen	1033
20.4.5 Meldung als E-Mail versenden	1033
20.4.6 Meldungen in JMS integrieren	1035
 Index	 1041