



***Oracle PL/SQL für Experten - Performance Analyse und Laufzeitoptimierung***  
***Seminarunterlage***  
***Version: 12.12***

Dieses Dokument wird durch die ORDIX AG veröffentlicht.

Copyright ORDIX AG. Alle Rechte vorbehalten.

Alle Produkt- und Dienstleistungs-Bezeichnungen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Firmen und beziehen sich auf Eintragungen in den USA oder USA-Warenzeichen.

Weitere Logos und Produkt- oder Handelsnamen sind eingetragene Warenzeichen oder Warenzeichen der jeweiligen Unternehmen.

Kein Teil dieser Dokumentation darf ohne vorherige schriftliche Genehmigung der ORDIX AG weitergegeben oder benutzt werden.

### **Adressen der ORDIX AG**

Die ORDIX AG besitzt folgende Geschäftsstellen

ORDIX AG  
Karl-Schurz-Str. 19a  
D-33100 Paderborn  
Tel.: (+49) 0 52 51 / 10 63 - 0  
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG  
An der alten Ziegelei 5  
D-48157 Münster  
Tel.: (+49) 02 51 / 9 24 35 – 00  
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG  
Welser Straße 9  
D-86368 Gersthofen  
Tel.: (+49) 08 21 / 507 492 – 0  
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG  
Kreuzberger Ring 13  
D-65205 Wiesbaden  
Tel.: (+49) 06 11 / 7 78 40 – 00  
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG  
Wikingerstraße 18-20  
D-51107 Köln  
Tel.: (+49) 02 21 / 8 70 61 – 0  
Fax.: (+49) 01 80 / 1 67 34 90

ORDIX AG  
Südwestpark 67/2  
D-90449 Nürnberg  
Tel.: (+49) 0 52 51 / 10 63 - 0  
Fax.: (+49) 01 80 / 1 67 34 90

Internet: <http://www.ordix.de>

Email: [seminare@ordix.de](mailto:seminare@ordix.de)

# Inhaltsverzeichnis

<b>1</b>	<b>Datentypen</b>	<b>7</b>
1.1	Ganzzahlen	8
1.1.1	PLS_INTEGER	8
1.1.2	SIMPLE_INTEGER	9
1.1.3	Überlauf SIMPLE_INTEGER	10
1.1.4	Performancevergleich	11
1.2	Fließkommazahlen (BINARY_FLOAT, BINARY_DOUBLE)	13
1.2.1	Fließkommazahlen Übersicht	13
1.2.2	BINARY_FLOAT und BINARY_DOUBLE	14
1.2.3	Beispiel mit BINARY_FLOAT und BINARY_DOUBLE	15
1.2.4	BINARY_FLOAT und BINARY_DOUBLE Konstanten	18
1.2.5	BINARY_FLOAT Datentyp	20
1.2.6	SIMPLE_DOUBLE Datentyp	21
1.2.7	Beispiel mit SIMPLE_FLOAT und SIMPLE_DOUBLE	22
1.2.8	Performancevergleich	24
1.3	VARCHAR2 Variablenlänge >4000	25
1.3.1	Beispiel mit VARCHAR2 Variablenlänge >4000	26
1.4	Übungen	28
<b>2</b>	<b>Coding</b>	<b>29</b>
2.1	Context-Wechsel bei SQL und PL/SQL	30
2.1.1	SQL und PL/SQL - Vergleich	30
2.1.2	Context-Wechsel SQL und PL/SQL	32
2.1.3	Beispiel mit Context-Wechsel SQL und PL/SQL	33
2.1.4	WITH-PL/SQL-Klausel	34
2.1.5	Beispiel mit WITH-PL/SQL-Klausel	35
2.1.6	PRAGMA UDF	36
2.2	Bulk Collection & FORALL	37
2.2.1	Array Processing mit Bulk Collection & FORALL	37
2.2.2	Bulk Collect mit mehreren Collections	38
2.2.3	Bulk Collect mit einer Collection	39
2.2.4	FORALL	40
2.2.5	FORALL mit RETURNING-Klausel	41
2.2.6	FORALL SAVE EXCEPTIONS	42
2.2.7	Beispiel mit FORALL und SAVE EXCEPTIONS	43
2.3	Logische Test mit dem günstigsten Vergleich zuerst	44
2.3.1	Boolesche Algebra	44
2.3.2	OR – Verknüpfung	45
2.3.3	AND – Verknüpfung	46
2.3.4	CASE – Anweisung	47
2.4	OUT-Parameter und NOCOPY-Hint	48
2.4.1	NOCOPY Compiler Hint Überblick	48
2.4.2	NOCOPY Compiler Hint Verwendung	49
2.4.3	NOCOPY Compiler Hint Parametermodi	50
2.4.4	NOCOPY Compiler Hint - Hinweise	51
2.4.5	NOCOPY Compiler Hint – Einschränkungen	52
2.5	Gruppieren von Funktionen/Prozeduren in Packages	54
2.5.1	Gruppierung Übersicht	54
2.5.2	Vorteile der Gruppierung	55
2.6	Festhalten der DB-Objekte im Shared Pool mit dbms_shared_pool	57
2.6.1	DBMS_SHARED_POOL Übersicht	57
2.6.2	DBMS_SHARED_POOL Beispiel	58
2.6.3	DBMS_SHARED_POOL Verwendungshinweise	60
2.7	Compound Trigger	61
2.7.1	Compound Trigger – Merkmale	61
2.7.2	Compound Trigger – Zeitpunkte	62
2.7.3	Compound Trigger Deklaration	63

2.7.4	Beispiel ohne Compound Trigger.....	64
2.7.5	Beispiel mit Compound Trigger.....	65
2.7.6	Compound Trigger Performancevorteile.....	67
2.8	LOB Tuning.....	68
2.8.1	Erzeugen von LOB SecureFiles.....	68
2.8.2	Aktivierung LOB SecureFiles.....	69
2.8.3	Einsatzmöglichkeiten.....	70
2.8.4	Komprimierung bei SecureFiles.....	71
2.8.5	Beispiele für Komprimierung bei SecureFiles.....	72
2.8.6	Deduplication bei SecureFiles.....	73
2.8.7	Beispiele für Deduplication bei SecureFiles.....	74
2.8.8	Transparente Verschlüsselung bei SecureFiles (TDE).....	75
2.8.9	Beispiele für Verschlüsselung bei SecureFiles.....	76
2.8.10	Inline versus out-of-line Speicherung.....	77
2.8.11	Performancevergleich BasicFiles und SecureFiles.....	79
2.8.12	Empfehlungen zum Setup von LOBs.....	80
2.9	PL/SQL Function Result Cache.....	81
2.9.1	Deterministische Funktionen.....	81
2.9.2	Result Cache.....	82
2.9.3	Server Result Cache Überblick.....	83
2.9.4	Monitoring des Server Result Cache.....	85
2.9.5	PL/SQL-Package DBMS_RESULT_CACHE.....	86
2.9.6	Hinweise.....	88
2.9.7	PL/SQL Function Result Cache.....	90
2.10	REGEXP-Funktionen.....	94
2.10.1	REGEXP – Einleitung.....	94
2.10.2	REGEXP – Anwendung.....	96
2.10.3	REGEXP – Funktionen.....	100
2.10.4	REGEXP – Metazeichen.....	102
2.10.5	REGEXP – Metaklassen.....	103
2.10.6	REGEXP – NLS.....	105
2.10.7	REGEXP – Performance.....	106
2.11	Collections.....	108
2.11.1	Collections Übersicht.....	108
2.11.2	Collections Eigenschaften.....	109
2.11.3	Collections Syntax.....	110
2.11.4	Funktionen für Collections.....	111
2.11.5	Ausnahmebehandlung der Collections.....	113
2.11.6	Unterschied zwischen DELETE und TRIM am Beispiel von Nested Table.....	115
2.11.7	Performancevergleich Collections.....	116
2.12	Multi-Table-Insert.....	118
2.12.1	Multi-Table-Insert Syntax.....	118
2.12.2	Multi-Table-Insert Beispiel.....	120
2.13	Bulk-Update.....	121
2.13.1	Bulk-Update Syntax.....	121
2.14	MERGE.....	122
2.14.1	MERGE Syntax.....	122
2.14.2	MERGE Beispiel.....	124
2.14.3	Performancevergleich MERGE.....	125
2.15	SQL Pattern Matching.....	128
2.15.1	MATCH_RECOGNIZE Übersicht.....	128
2.15.2	MATCH_RECOGNIZE Syntax.....	129
2.15.3	MATCH_RECOGNIZE Syntax I.....	130
2.15.4	MATCH_RECOGNIZE Syntax II.....	131
2.15.5	MATCH_RECOGNIZE Syntax III.....	132
2.15.6	MATCH_RECOGNIZE Beispiel I.....	133
2.15.7	MATCH_RECOGNIZE Beispiel II.....	134
2.16	Parallele SQL-Verarbeitung.....	135
2.16.1	Parallele SQL-Verarbeitung Übersicht.....	135

2.16.2	Parallelisierbare SQL-Befehle .....	136
2.16.3	Aktivierung der parallelen Verarbeitung .....	138
2.16.4	Besonderheiten bei der parallelen Verarbeitung .....	139
2.16.5	INIT.ORA Parameter PARALLEL_DEGREE_POLICY .....	140
2.16.6	Automatische Degree of Parallelism (AUTO DOP) .....	142
2.16.7	Parallel Statement Queuing .....	144
2.16.8	In-Memory Parallel Execution .....	145
2.17	Pipelined Table Function .....	146
2.17.1	Pipelined Table Function Syntax .....	146
2.17.2	Pipelined Table Function Beispiel .....	147
2.18	Übungen .....	149
<b>3</b>	<b>Compiler-Optionen .....</b>	<b>155</b>
3.1	Native Kompilierung .....	156
3.1.1	Vergleich PLSQL_CODE_TYPE INTERPRETED/NATIVE .....	156
3.1.2	Funktionsweise Native Kompilierung .....	157
3.1.3	Vorteile Native Kompilierung .....	158
3.1.4	Beispiel mit Native Kompilierung .....	159
3.1.5	Native Kompilierung der gesamten Datenbank .....	160
3.2	Subprogram inlining .....	161
3.2.1	Was ist Subprogram inlining? .....	161
3.2.2	Beispiel mit Subprogram inlining .....	162
3.2.3	Initialisierungsparameter PLSQL_OPTIMIZE_LEVEL .....	163
3.2.4	PRAGMA INLINE .....	164
3.2.5	Beispiel mit PRAGMA INLINE .....	165
3.3	PL/SQL Warnungen .....	166
3.3.1	Übersicht PL/SQL-Compilerwarnungen .....	166
3.3.2	Funktionsweise von PLSQL_WARNINGS .....	167
3.3.3	Syntax von PLSQL_WARNINGS .....	168
3.3.4	Richtlinien für PLSQL_WARNINGS .....	169
3.3.5	Beispiel mit PLSQL_WARNINGS .....	170
3.3.6	PL/SQL-Package DBMS_WARNING .....	173
3.3.7	Beispiel mit DBMS_WARNING .....	175
3.4	Übungen .....	176
<b>4</b>	<b>XMLType .....</b>	<b>178</b>
4.1	Überblick XMLType .....	179
4.2	Vorstellung der einzelnen Speicherungsformen .....	180
4.2.1	Unstrukturierte Speicherung als CLOB .....	181
4.2.1.1	Create mit XMLType .....	182
4.2.2	Strukturierte Speicherung in objektrelationalen Tabellen .....	183
4.2.2.1	Beispiel mit Strukturierter Speicherung .....	184
4.2.3	Speicherung als Binary XMLType .....	185
4.2.3.1	Vorteile von Binary XML .....	186
4.2.3.2	Anlegen einer Tabelle .....	187
4.2.3.3	Encoding Optionen beim Create Table .....	188
4.3	Wahl der Speicherungsform .....	193
4.3.1	Unstrukturierte Speicherung .....	194
4.3.2	Objektrelationale Speicherung .....	195
4.3.3	Binary XMLType .....	196
4.3.4	Vergleich der Laufzeiten .....	197
4.4	Indizierung .....	200
4.4.1	Indizierung bei Objektrelationaler Speicherung .....	200
4.4.1.1	Anlegen eines Indexes .....	201
4.4.2	Indizierung bei Unstrukturierte Speicherung und Binary XMLType .....	202
4.4.2.1	Aufbau der Path-Table .....	204
4.4.2.2	Vorteile von XMLIndex .....	205
4.4.2.3	Nicht indizierbare XPath-Ausdrücke .....	206
4.4.2.4	Anlegen eines XMLIndex .....	207
4.4.2.5	Anlegen von secondary Indexes .....	208

---

4.5	Vorgehensweise bei Tuning und Abfragen .....	209
4.6	Übungen .....	210
<b>5</b>	<b>SQL Tuning .....</b>	<b>212</b>
5.1	SQL in PLSQL .....	213
5.1.1	Wie kann ich mein SQL überprüfen? .....	214
5.2	Tuningansätze für SQL-Befehle .....	216
5.3	Indexes .....	218
5.3.1	Indexes .....	218
5.4	Statistiken .....	222
5.5	Sortierung .....	224
5.6	Übungen .....	225
<b>6</b>	<b>Tools zu Performanceanalyse .....</b>	<b>226</b>
6.1	Hierarchischer Profiler .....	227
6.1.1	Hierarchischer Profiler Überblick .....	227
6.1.2	Hierarchischer Profiler .....	228
6.1.3	Features des Hierarchischen Profilers .....	229
6.1.4	Bestandteile .....	230
6.1.5	Fünf Schritte der Datensammlung .....	231
6.1.6	Hierarchischer Profiler Datenbanktabelle .....	233
6.1.7	PLSHPROF Utility .....	234
6.1.8	PLSHPROF HTML Report .....	235
6.2	Übungen .....	236
<b>7</b>	<b>Tipps und Tricks .....</b>	<b>237</b>
7.1	NOT NULL Constraint .....	238
7.2	ROWID .....	239
7.3	Error Logging Tabelle .....	240
7.3.1	DBMS_ERRLOG .....	240
7.3.2	Error Logging beim Insert .....	241
7.3.3	Error Logging Einschränkungen .....	242
7.4	NVL, COALESCE und CASE .....	243
7.5	PL/SQL execution elapsed time .....	244