

Section	Title	Page	
<b>One</b>	<b>Advanced PL/SQL 12c Introduction</b>	2	
	- Introduction	3	
	- Overview of PLSQL Structure	4	
	- Overview of DBMS_OUTPUT	5	
	- Overview of Exceptions	8	
	- Overview of raise_application_error	11	
	- Overview of Alternative Quoting Mechanism	13	
	- Overview of Sequences	15	
	- Overview of Savepoints	16	
	- Overview of Conditional Statements	18	
	<b>Two</b>	<b>Defining Variables in PLSQL</b>	25
		- Variables in PLSQL	26
		- Quoted Identifier Variables in PLSQL	27
- Numeric Variables in PLSQL		30	
- Number Variables in PLSQL		31	
- Decimal Variables in PLSQL		34	
- Integer Variables in PLSQL		35	
- PLS_Integer Variables in PLSQL		36	
- Binary_Integer Variables in PLSQL		37	
- Natural Variables in PLSQL		38	
- Naturaln Variables in PLSQL		39	
- Positive Variables in PLSQL		40	
- Positiven Variables in PLSQL		41	
- Signtype Variables in PLSQL		42	
- Simple_Integer Variables in PLSQL		43	
- Defining Variables Using %Type		44	
- Defining Variables Using %Rowtype		45	
- Populating a Variable which is a %Rowtype		47	
- Defining Constants		48	
<b>Three</b>		<b>Sub-blocking in PLSQL</b>	50
		- Introduction	51
		- Defining Sub-Blocks	52
		- Re-Raising Exceptions in PLSQL	57
<b>Four</b>		<b>Loop Processing in PLSQL</b>	60
		- Overview of Looping in PLSQL	61
		- Basic Looping in PLSQL	62
		- While Looping in PLSQL	63
	- For Looping in PLSQL	64	
	- Using Dates in For Loops in PLSQL	66	
	- Using Characters in For Loops in PLSQL	67	
	- Loop Labels in PLSQL	68	
	- Loop Continue and Continue When in PLSQL	70	
<b>Five</b>	<b>Timestamps</b>	74	
	- Timestamps and Intervals in PLSQL	75	
	- Timestamp Variables in PLSQL	76	
	- Timestamp in PLSQL	78	
	- Populating Timestamps in PLSQL	81	

Section	Title	Page
	- Using Extract in PLSQL	83
	- Timestamp with Time Zone	84
	- Timestamp Time Zone and Extract	87
	- Using sys_extract_utc to normalise Timestamps	88
	- Using the To_Timestamp_TZ function	89
	- Using the From_TZ function	91
	- Timestamp with Local Time Zone	93
	- Intervals	95
	- Populating Year to Month Intervals	98
	- Populating Day to Second Intervals	99
	- Using Intervals in PLSQL	100
	- Using Intervals with Extract in PLSQL	105
	- Populating Intervals using Timestamps	106
	- <i>Exercise One</i>	107
<b>Six</b>	<b>Case Statements and Case Expressions</b>	<b>113</b>
	- Introduction	114
	- Using Searched Case in PLSQL	115
	- Using Case in PLSQL	116
	- Using Nested Case in PLSQL	117
	- Using Case Expressions in PLSQL	119
	- Using Nullif in PLSQL	123
	- Using Coalesce in PLSQL	124
<b>Seven</b>	<b>Cursors in Oracle 12c</b>	<b>125</b>
	- Overview of Cursors	126
	- Overview of Implicit Cursors	127
	- Looping an Implicit Cursor with a For Loop	129
	- Overview of Explicit Cursors	130
	- Overview of Explicit Cursor Attributes	132
	- Overview of Explicit Cursor Parameters	133
	- Looping an Explicit Cursor with a For Loop	135
	- Using other Loops with Explicit Cursors	136
	- Explicit Cursors Versus Implicit Cursors	139
<b>Eight</b>	<b>Advanced Explicit Cursors</b>	<b>140</b>
	- For Update with Explicit Cursors	141
	- Where Current Of Explicit Cursors	142
	- Example of Updating with Explicit Cursors	143
	- Using Nowait with Explicit Cursors	144
	- Using Skip Locked with Explicit Cursors	145
	- Using Wait with an Explicit Cursor	146
	- Using Rowid with Explicit Cursors	147
	- Using Returning Into in PLSQL	148
	- Using Rowtype for DML	151
	- Populating a Variable which is a %Rowtype	153
	- Passing a %Rowtype between Processes	155
	- Declaring a Cursor %Rowtype	157
	- <i>Exercise Two</i>	160

Section	Title	Page
<b>Nine</b>	<b>Overview of PL/SQL Database Objects</b>	164
	- PLSQL Objects	165
	- Function syntax	166
	- Procedure syntax	167
	- Advantages of Packaging	168
	- Package syntax	169
	- Package Body syntax	171
	- Re-Compiling PLSQL Objects	172
	- Defining Parameters in PLSQL	173
	- Running Objects in PLSQL	176
	- Passing Parameters	179
	- Out Parameters in PLSQL	181
	<b>Ten</b>	<b>Advanced Object Techniques</b>
- Sub-programming		184
- Sub-programming Example		185
- Sub-programming Limitations		186
- Forward Declaration		187
- Autonomous Transactions		191
- Example of an Autonomous Transactions		193
- Ref Cursors in PLSQL		200
- Strongly Typed Ref Cursors in PLSQL		201
- Weakly Typed Ref Cursors in PLSQL		203
- Sys_refcursor in PLSQL		204
- Using Sys_Refcurors with SQL		206
- Private Processes in PLSQL Packages		210
- Handling Exceptions in Packages in PLSQL		213
<b>Eleven</b>		<b>Overloading Modules in Packages</b>
	- Introduction	216
	- Example	217
	- User_Procedures	222
<b>Twelve</b>	<b>Advanced Packages</b>	223
	- Introduction	224
	- Defining Explicit Cursors in Packages	225
	- Flexible Explicit Cursors in Packages	229
	- Passing Parameters Using Packages	232
	- Global Variables using Packages	235
	- Public and Private Variables in Packages	242
<b>Thirteen</b>	<b>Advanced Exception Handling</b>	244
	- The Third Boolean Parameter	245
	- Raising User Defined Exceptions in PLSQL	246
	- Pragma Exceptions in PLSQL	249
	- Alternative Way to Handle SQL Errors in PLSQL	251
	- Handling SQL Errors in PLSQL	252
	- Advanced SQLErrm	254
	- Nesting Exceptions in PLSQL	255
	- Re-Raising Exceptions in PLSQL	258
	- Handling Exceptions in Sub-Processes in PLSQL	260

Section	Title	Page
	- Using Nocopy When Declaring Processes	264
	- Using DBMS_UTILITY Functions	268
	- Using DBMS_UTILITY.FORMAT_ERROR_STACK	269
	- Using DBMS_UTILITY.FORMAT_BACKTRACE	270
	- <i>Exercise Three</i>	273
<b>Fourteen</b>	<b>DML Triggers in Oracle</b>	<b>284</b>
	- Introduction	285
	- DML Triggers Introduction	286
	- DML Triggers Timing Points	289
	- DML Triggers When Conditions	290
	- DML Triggers When Conditions Examples	291
	- Syntax of DML Triggers in PLSQL	293
	- Referencing Values within Triggers	294
	- Checking the Event which fired the DML Trigger	298
	- Using the Follows syntax to dictate Trigger Firing	300
	- Using the Disable syntax when Compiling Triggers	305
	- Enabling/Disabling Triggers in an Oracle Database	306
	- Metadata for Triggers (User_Triggers)	308
	- Mutating DML Triggers in Oracle	309
	- Using Autonomous Transactions in Triggers	313
<b>Fifteen</b>	<b>Instead Of Triggers</b>	<b>315</b>
	- Introduction to Instead Of Triggers	316
	- Syntax of an Instead Of Trigger in Oracle	319
	- Example of an Instead Of Trigger	321
	- Deleting Records in Oracle using an Instead Of	329
	- Using Follows in an Instead Of Trigger	330
	- <i>Exercise Four</i>	331
<b>Sixteen</b>	<b>Compound Triggers</b>	<b>340</b>
	- Introduction to Compound Triggers	341
	- Syntax for Table Compound Triggers	343
	- Declaration Section	344
	- Timing Points	345
	- BEFORE STATEMENT	346
	- BEFORE EACH ROW	347
	- AFTER EACH ROW	348
	- AFTER STATEMENT	349
	- Compound Triggers for Views	350
<b>Seventeen</b>	<b>System Triggers</b>	<b>351</b>
	- Introduction to System Triggers	352
	- Syntax for System Triggers in Oracle	353
	- System Event Triggers	354
	- User Event Triggers	355
	- Schema Event Triggers	359
	- Database Level Attributes for System Triggers	365
	- <i>Exercise Five</i>	366

<b>Section</b>	<b>Title</b>	<b>Page</b>
<b>Eighteen</b>	<b>Using Contexts</b>	371
	- Introduction	372
	- Attributes	373
	- User Defined	381
	- Creating Contexts	382
	- Creating a Package for a Context	383
	- Creating a Context	384
	- Populating a Context	385
	- Accessing a Context	386
	- Listing Contents of a Context	387
	- Deleting Contents of a Context	389
	- Changing Context Behaviour	391
<b>Nineteen</b>	<b>Subprogram Inlining</b>	392
	- Introduction	393
	- Pragma Inline	394
	- Example of Inlining	395
	- Inlining Usage	400
	- Conclusion	402
<b>Twenty</b>	<b>PLSQL Compiler</b>	403
	- Introduction	404
	- Warning Levels	405
	- Enabling/Disabling Warning Levels	406
	- Using the Alter Session	407
	- Using the Error option	410
	- Using DBMS_WARNING Package	412
	- Using DBMS_WARNING.GET_CATEGORY	413
	- Severe Category Errors	414
	- Informational Category Errors	416
	- Performance Category Errors	418
	- Using dbms_warning.set_warning_setting_string	419
	- Using dbms_warning.set_warning_setting_num	420
	- Using dbms_warning.set_warning_setting_cat	421
	- USER_PLSQL_OBJECT_SETTINGS	422
	- Re-Compiling with same Warning settings	424
	- PLSQL_OPTIMIZE_LEVEL	425
	- Optimize Levels	426
	- Timing Optimization with dbms_utility.get_cpu_time	427
	- Optimize Levels in user_plsql_object_settings	433
	- Re-Compiling with same Optimize settings	434
	- High Level Optimization without Pragma Inline	435
<b>Twenty One</b>	<b>Conditional Compilation in PLSQL</b>	437
	- Introduction	438
	- Dbms_preprocessor	440
	- Compiling using PLSQL_CCFLAGS	441
	- PLSQL_CCFLAGS in PLSQL	443
	- Using \$error Directive	451
	- Using Constants with CCFlags	453

Section	Title	Page
	- \$\$PLSQL_CODE_TYPE	454
	- Oracle Compilation Modes	455
	- \$\$PLSQL_OPTIMIZE_LEVEL	456
	- \$\$PLSQL_LINE	457
	- \$\$PLSQL_UNIT	458
	- DBMS_DB_VERSION	459
	- Setting Session CCFlags	461
	- Plsql_ccflags in user_plsql_object_settings	462
	- Plsql_ccflags in Triggers	463
<b>Twenty Two</b>	<b>Regular Expressions in PLSQL (Regex)</b>	465
	- Introduction	466
	- Using REGEXP_LIKE in Where Clauses	467
	- Case Sensitivity	469
	- Line Anchors	470
	- Using the Dot (Any Character)	472
	- Counting Characters {Intervals}	475
	- Character Lists	479
	- Excluding Character Lists	483
	- Alternatives to Character Lists	484
	- Class Shorthands	487
	- Or Patterns (Alternatives)	488
	- Or Patterns with Character Lists	489
	- Using the Question Mark (Optional)	490
	- Using the Plus sign (Mandatory)	493
	- Using the Star sign (Optional)	495
	- Backreferencing	496
	- POSIX	498
	- Using POSIX in Oracle - [:upper] example	500
	- Using POSIX in Oracle - [:digit] example	501
	- Using POSIX in Oracle – Character equivalents	502
	- Using Regexp_like in PLSQL	503
	- Using Regexp_count in PLSQL	506
	- Using Regexp_replace in PLSQL	510
	- Using Regexp_instr in PLSQL	512
	- Using Regexp_substr in PLSQL	514
	- Using Regular Expressions in Oracle	516
	- <i>Exercise Six</i>	517
<b>Twenty Three</b>	<b>File I/O</b>	524
	- Reading and Writing from the Operating System in PLSQL	525
	- Accessing the File System using Directories	526
	- Creating Directories	527
	- Simple Write using UTL_FILE.PUT_LINE	529
	- Simple Read using UTL_FILE.GET_LINE	530
	- Additional Functions in the UTL_FILE Package	531
	- Using UTL_FILE.PUTF in PLSQL	532
	- UTL_FILE Exceptions	534

Section	Title	Page
	- Using UTL_FILE to perform Operating System Commands	536
	- Using UTL_FILE.Fcopy in PLSQL	537
	- Using UTL_FILE.Fgetattr in PLSQL	538
	- Using UTL_FILE.Fremove in PLSQL	539
	- Using UTL_FILE.Frename in PLSQL	540
<b>Twenty Four</b>	<b>Working with Clobs and Blobs in PLSQL</b>	<b>541</b>
	- Introduction	542
	- Creation of Clobs in Tables	543
	- Using Clobs in PLSQL	544
	- Inserting Clobs into the Database using PLSQL	545
	- Writing Clobs to the Filesystem using PLSQL	549
	- Introduction to Blobs	550
	- Inserting Blobs into the Database using PLSQL	551
	- Writing Blobs into the Filesystem using PLSQL	552
	- DBMS_LOB exceptions	555
<b>Twenty Five</b>	<b>DBMS_Metadata</b>	<b>557</b>
	- Generating Database Object Scripts	558
	- List of Object Types	559
	- Example of Using DBMS_Metadata	560
<b>Twenty Six</b>	<b>Encrypting Code</b>	<b>562</b>
	- Introduction	563
	- Using Wrap.exe to encrypt processes in Oracle	564
	- Example of using Wrap.exe to encrypt processes	567
	- Using DBMS_DDL.CREATE_WRAPPED Function	569
	- <i>Exercise Seven</i>	573
<b>Twenty Seven</b>	<b>Object Orientated Programming</b>	<b>577</b>
	- Introduction	578
	- Definition of Objects	579
<b>Twenty Eight</b>	<b>Row Objects</b>	<b>585</b>
	- Introduction	586
	- Creating	587
	- Metadata	589
	- Data	590
	- Indexes	591
	- Views	593
	- Removing	594
	- OIDs	596
<b>Twenty Nine</b>	<b>Column Objects</b>	<b>604</b>
	- Introduction	605
	- Describing	606
	- Inserting into	609
	- Selecting from	611
	- Updating	617
	- Object Views	618
<b>Thirty</b>	<b>Defining Processes within Objects</b>	<b>620</b>
	- Introduction	621
	- Defining Methods	622

Section	Title	Page
	- Member Methods	624
	- Map Methods	631
	- Order Methods	633
	- Constructor Methods	639
<b>Thirty One</b>	<b>PLSQL Data Structures and Collections</b>	643
	- Programmer Defined Records	644
	- Nested Programmer Defined Records	648
	- Varrays in PLSQL	650
	- Using Varrays in Loops	653
	- Populating Varrays	654
	- Extending Varrays	655
	- Deleting from Varrays	657
	- Varrays and Tables	658
	- Multilevel Varrays	661
<b>Thirty Two</b>	<b>Associate Arrays (Index by Tables)</b>	664
	- Introducing Associate Arrays in PLSQL	665
	- Declaring PLSQL Tables	667
	- Populating PLSQL Tables	668
	- Using Count with PLSQL Tables	670
	- Using Delete with PLSQL Tables	671
	- Using Exists with PLSQL Tables	672
	- Using First/Last with PLSQL Tables	673
	- Using Next/Prior with PLSQL Tables	674
	- Using Varchar2 as an index with a PLSQL Table	675
	- Multi-Level PLSQL Table	679
	- Nested PLSQL Table	680
	- Cardinality with Nested Table	683
	- Tidying Nested PLSQL Tables using Set	684
	- Nested PLSQL Tables using Set	685
	- Trimming Nested PLSQL Tables	687
	- Multi-Level Nested PLSQL Tables	689
	- Using PLSQL Tables as Parameters	690
<b>Thirty Three</b>	<b>Collection Comparisons</b>	694
	- Introduction	695
	- Check for Equality	696
	- Using IN with a Nested Table	699
	- Using Member Of with a Nested Table	700
	- Using Is Empty with a Nested Table	701
	- Multiset Union	702
	- Multiset Union Distinct	704
	- Multiset Intersect	705
	- Multiset Except	706
	- Submultiset	707
	- Not Submultiset	708
	- Not Submultiset Alternative	709
	- <i>Exercise Eight</i>	710



Section	Title	Page
<b>Thirty Four</b>	<b>Bulk SQL in PLSQL</b>	713
	- Introduction to Bulk Processing	714
	- Bulk Binding in PLSQL	715
	- Bulk Binding in PLSQL Using Associate Arrays	716
	- Bulk Binding in PLSQL Using Varrays	717
	- Bulk Binding in PLSQL Using Nested Tables	718
	- Forall with Save Exceptions	719
	- Forall with SQL%BULKEXCEPTIONS	721
	- Forall with Indices Of	724
	- Forall with Values Of	726
	- Bulk Collect with Implicit Cursors (Varray)	727
	- Bulk Collect with Implicit Cursors (Associative Array)	728
	- Bulk Collect with Implicit Cursors (Nested Table)	729
	- Bulk Collect with Explicit Cursors	730
	- Bulk Collect and Returning in PLSQL	731
	- Problems with Bulk Processing in PLSQL	734
	- Bulk Processing and Limit in PLSQL	735
<b>Thirty Five</b>	<b>Invoker Rights in PLSQL</b>	736
	- The need to use Invoker Rights	737
	- Example of Invoker Rights	738
	- Using AUTHID in PLSQL	741
<b>Thirty Six</b>	<b>Virtual Private Databases (VPD)</b>	744
	- Introduction	745
	- DBMS_RLS	746
	- Creating Functions for use with VPD	747
	- Applying Policies	749
	- Viewing Policies in Metadata	752
	- Row Level Security (RLS)	753
	- Removing Row Level Security (RLS)	754
	- Using Policy Types with DBMS_RLS	756
	- Introducing Flexibility to VPD	757
	- Column Level Masks	758
	- Using Sec_relevant_cols	759
	- Using Sec_relevant_cols_opt	762
	- Introducing Flexibility to VPD	757
<b>Thirty Seven</b>	<b>Dynamic SQL</b>	765
	- Introduction to Dynamic SQL	766
	- Native Dynamic SQL (NDS) with Execute Immediate	767
	- NDS Error Handling	770
	- NDS with Inputs	771
	- NDS with Output	773
	- Dynamic Cursors and Sys_Refcursor	774
	- NDS with Outputs	776
	- Introduction to DBMS_SQL	777
	- DBMS_SQL.Example	778
	- <i>Exercise Nine</i>	779