DATABASES INTERMEDIATE 4 DAYS

Oracle PL/SQL Programming

Four days of PL/SQL for developers building and maintaining Oracle application logic. Covers language fundamentals, cursors, exceptions, procedures, functions and packages, collections, bulk processing with BULK COLLECT and FORALL, triggers, dynamic SQL, JSON and LOBs, compilation and dependencies, performance tuning and secure coding. Delivered on Oracle 19c and 23ai with labs on a realistic schema.

Why Oracle PL/SQL Training?

Bulk processing changes the numbers

Converting row-by-row loops to BULK COLLECT and FORALL routinely cuts runtimes by an order of magnitude, and it gets a dedicated module.

Injection-safe dynamic SQL

Native dynamic SQL and DBMS_SQL are taught with bind variables and DBMS_ASSERT rather than avoided.

Packages as the unit of design

Teams learn to structure code into packages with clear interfaces rather than accumulating standalone procedures.

Debugging and testing included

Instrumentation, profiling and unit testing are part of the course, so code arrives maintainable rather than merely working.

What you'll be able to do

- Write structured PL/SQL blocks using appropriate types and control flow
- Embed SQL in PL/SQL and use implicit and explicit cursors correctly
- Implement layered exception handling with meaningful error contracts
- Build procedures, functions and packages with clean interfaces
- Use collections and records to process structured data in memory
- Apply BULK COLLECT and FORALL to eliminate context switching overhead
- Implement triggers appropriately and avoid mutating table problems
- Write injection-safe dynamic SQL using bind variables
- Work with JSON, large objects and file handling from PL/SQL
- Manage dependencies, invalidation and conditional compilation
- Profile and tune PL/SQL for measurable performance improvement
- Instrument, debug, unit test and secure production PL/SQL code

Who this is for

- Application developers working against Oracle databases
- Database developers and application DBAs
- Developers migrating to Oracle from another platform
- Data engineers building Oracle based ETL logic
- Technical leads setting PL/SQL coding standards

Curriculum

01 PL/SQL Language Foundations

- Block structure: declaration, execution and exception sections
- Anonymous blocks, named units and where each belongs
- How PL/SQL and SQL engines interact and the cost of context switching
- Development environment setup and compilation feedback
- Coding standards, naming and formatting conventions
- **Hands-on:** build and compile a first set of blocks with deliberate errors to diagnose

02 Types, Variables and Control Structures

- Scalar types, anchored declarations with %TYPE and %ROWTYPE
- Subtypes, constants and variable scope and visibility
- Conditional logic with IF and CASE
- Loop constructs and loop control statements
- Nested blocks, labels and scope resolution
- **Hands-on:** implement a business rules routine with layered control flow

03 SQL Within PL/SQL

- SELECT INTO and handling no data found and too many rows
- DML statements and implicit cursor attributes
- Transaction control: COMMIT, ROLLBACK and savepoints
- Autonomous transactions and their appropriate use
- The RETURNING clause for capturing modified values
- **Hands-on:** build a transactional routine with correct commit boundaries

04 Cursors and Result Set Processing

- Explicit cursor declaration, opening, fetching and closing
- Cursor FOR loops and parameterised cursors
- Cursor variables, REF CURSOR and returning result sets to clients
- FOR UPDATE and WHERE CURRENT OF for locked processing
- Choosing between cursor loops and set-based SQL
- **Hands-on:** return a result set to a calling application via a REF CURSOR

05 Exception Handling

- Predefined, non-predefined and user-defined exceptions
- RAISE_APPLICATION_ERROR and designing error codes
- SQLCODE, SQLERRM and DBMS_UTILITY error stack functions
- Exception propagation across nested blocks and program units
- Logging errors without swallowing them
- **Hands-on:** implement an error handling and logging framework

06 Procedures and Functions

- Parameter modes, defaults and named notation
- Functions in SQL statements and the purity requirements
- Deterministic functions and result cache
- Invoker rights versus definer rights
- Overloading, forward declaration and recursion
- **Hands-on:** build a set of stored program units with documented interfaces

07 Packages

- Specification and body separation as an interface contract
- Public and private constructs and information hiding
- Package state, initialisation blocks and session persistence
- Overloading within packages and cursor variable exposure
- Useful supplied packages: DBMS_OUTPUT, UTL_FILE, DBMS_SCHEDULER, DBMS_LOB
- **Hands-on:** refactor standalone procedures into a coherent package API

08 Collections and Records

- Associative arrays, nested tables and VARRAYs compared
- Records, nested records and record based DML
- Collection methods, iteration and sparse collections
- Multilevel collections and collections of records
- Persistent collection types in the database
- **Hands-on:** process a complex dataset in memory using collections

09 Bulk Processing and Performance Patterns

- BULK COLLECT with LIMIT for controlled memory use
- FORALL for bulk DML and SAVE EXCEPTIONS for partial failure
- Measuring the context switch cost you eliminate
- Pipelined table functions and parallel enabled functions
- When set-based SQL still beats any PL/SQL approach
- **Hands-on:** convert a row-by-row routine to bulk processing and benchmark both

10 Triggers

- DML trigger types, timing and row versus statement level
- Compound triggers and resolving mutating table errors
- INSTEAD OF triggers on views
- System and DDL event triggers
- Trigger ordering, FOLLOWS and maintainability concerns
- **Hands-on:** implement an audit and validation trigger set without mutating errors

11 Dynamic SQL

- EXECUTE IMMEDIATE and native dynamic SQL
- Bind variables, USING and INTO clauses
- DBMS_SQL for statements with unknown structure at compile time
- SQL injection in PL/SQL and defending with DBMS_ASSERT
- Performance implications of dynamic versus static SQL
- **Hands-on:** build a dynamic query routine and prove it resists injection

12 JSON, Large Objects and External Data

- JSON storage, generation and querying from PL/SQL
- JSON_TABLE, path expressions and PL/SQL JSON object types
- CLOB, BLOB and BFILE handling with DBMS_LOB
- UTL_FILE for server side file reading and writing
- Calling web services and handling structured responses
- **Hands-on:** parse a JSON payload into relational tables

13 Dependencies, Compilation and Code Management

- Dependency tracking, invalidation and recompilation
- Fine-grained dependency behaviour and its effect on deployment
- Conditional compilation for version specific code
- Compiler warnings, PLSQL_WARNINGS and treating warnings as defects
- Native compilation and optimisation levels
- **Hands-on:** diagnose and resolve an invalidation cascade

14 Profiling and Tuning PL/SQL

- DBMS_PROFILER and DBMS_HPROF for hierarchical profiling
- Identifying context switching and excessive SQL execution
- NOCOPY hints, parameter passing and memory behaviour
- Caching strategies: result cache, deterministic functions and package variables
- Reducing PGA consumption in high concurrency workloads
- **Hands-on:** profile a slow package and deliver a measured improvement

15 Debugging, Testing and Secure Coding

- Instrumentation and application tracing with DBMS_APPLICATION_INFO
- Interactive debugging in modern development tools
- Unit testing PL/SQL with utPLSQL and building a regression suite
- Least privilege execution and code security review
- Wrapping, source protection and deployment practice
- Code review checklist for production PL/SQL
- **Hands-on:** capstone package delivered with tests, instrumentation and a review