



NYOUG Fall 2013 Training Session
**“High Performance PL/SQL:
Unleashing the Power of Server-Side Development”**
Presented by Michael Rosenblum, Dulcian, Inc.

DATE: November 7, 2014

TIME: 9:00 AM - 5:00 PM

LOCATION: BMCC 199 Chambers St. New York, NY 10007

SCHEDULE: (Break times are approximate)

Time	Activity	Location
8:00-9:00AM	Registration and Breakfast	
9:00-10:00 AM	Know Your Environment	
10:00-11:00 AM	User Defined PL/SQL Functions in the SQL Context	
11:00-11:15	BREAK	
11:15AM-12:15PM	SQL Sets Within PL/SQL	
12:15-1:15 PM	LUNCH	
1:15-2:15 PM	Triggers are NOT Evil	
2:15-3:15 PM	Keeping the Cache	
3:15-3:30 PM	BREAK	
3:30-4:30	WHEN OTHERS THEN...	
4:30-5:00PM	Questions	

Michael Rosenblum is a Software Architect/Senior DBA at Dulcian, Inc., where he is responsible for system tuning and application architecture. Michael supports Dulcian developers by writing complex PL/SQL routines and researching new features. He is the co-author of *Oracle PL/SQL Performing Tuning Tips & Techniques* (Oracle Press, 2014) *PL/SQL for Dummies* (Wiley Press, 2006), contributing author of *Expert PL/SQL Practices* (Apress, 2011), as well as the author of a number of database-related articles (appearing in *IOUG SELECT Journal* and *ODTUG Technical Journal*) and conference papers. Michael is an Oracle ACE, a frequent presenter at various Oracle user group conferences (Oracle OpenWorld, ODTUG, IOUG Collaborate, RMOUG, NYOUG, etc.), and winner of the ODTUG Kaleidoscope 2009 Best Speaker Award. In his native Ukraine, he graduated summa cum laude from the Kiev National Economic University, where he received a Master of Science degree in Information Systems.

Detailed Description of Session

Many developers (particularly those in the Java world), tend to minimize the use of the database. Some go so far as to place no code at all in the database and declare that database independence is a highly desirable goal. These developers are missing the most productive tool in their arsenal. PL/SQL code is tightly coupled with the Oracle database. It is vastly more efficient than any other development environment for processing data. Code in the database tends to execute a full order of magnitude faster than any other code (JavaEE or .Net) residing in the application server. Routines written in PL/SQL require up to 60% less code than other programming languages to perform exactly the same functions. In addition, when using server-side code, there is far less network traffic between the application server and the database.

The goal of this seminar is to illustrate how PL/SQL should be properly utilized in a real-world development environment. From database configuration settings to the nuances of caching techniques, it covers the entire depth and efficiency of PL/SQL-based solutions.

PRE-REQUISITES

1. Know your Environment – This section introduces the available tools and features related to PL/SQL performance tuning including: data dictionary views, logging, tracing, profiling etc. It also discusses the proper ways to instrument code deal with and exception handling.
2. User-Defined PL/SQL Functions in the SQL Context – For most developers, knowledge of PL/SQL starts from writing user-defined functions. As a result, even if this code is functionally correct, the program units are fired significantly more often than needed, impact CBO decisions, and cause execution plan degradation. This section addresses these issues and includes a number of examples of how PL/SQL can extend basic SQL functionality.
3. SQL Sets Within PL/SQL – The goal of PL/SQL is to efficiently communicate with SQL sets using cursors. This key feature of database cursors is often missed by assuming that they are nothing more than pointers to SQL sets. The most efficient way of working with SQL sets is to access data using set-based operators. This section reviews the results of two case studies focused on comparing different access approaches. The first one looks for the most efficient ways of moving significant volumes of data between remote locations, while the second tests the impact of MULTISET operations on resource utilization.
4. “Triggers are NOT Evil” - Both table triggers and INSTEAD-OF triggers are examined from a global system optimization point of view that includes not only the aspect of functional correctness, but also the tradeoffs between multiple goals. For example, depending upon the available hardware, developers can select either CPU-intense or I/O-intense solutions. This section also covers some of the most common performance problems related to different kinds of DML triggers and the proper ways of resolving them.
5. “Keeping the Cache” - One of the most important performance tuning strategies is to ensure that you are not doing exactly the same thing multiple times. However, often you cannot avoid this repetition. In those cases, the best option is to minimize the impact by utilizing existing caching techniques, either manual or built-in. Currently in Oracle, there are different caching mechanisms, each with its own strengths, drawbacks, and side-effects. This section introduces all of them and details the selection criteria for matching the appropriate mechanism to specific situations.

6. “WHEN OTHERS THEN...” – The topic of PL/SQL performance tuning is too large to cover in a 6-hour seminar. However, some tips and tricks can be very useful to attendees. This last section describes some one-off cases that overlap multiple areas. It also includes an extended question & answer period where attendees will get a chance to share their experiences and ask questions.

Outline

1. Introduction
2. Setting up the environment
 - a. PL/SQL-related database configuration
 - b. Oracle built-in tools
 - i. Logging/Tracing
 - ii. Hierarchical Profiler
 - c. Code instrumentation
3. User-defined PL/SQL functions in SQL context
 - a. Statistics management
 - i. Costing
 - ii. Selectivity
 - iii. Cardinality
 - b. Understanding call frequencies
 - c. Extending SQL functionality:
 - i. Custom Aggregate Functions
 - ii. Solving SQL problems procedurally
4. SQL sets within PL/SQL
 - a. Fetching mechanisms
 - b. Processing large data sets
 - c. Continuous page vs. pagination
 - d. Merging sets
5. Triggers
 - a. Table triggers
 - i. Data quality enforcement case study: Constraints vs. Triggers
 - ii. Default value management
 - iii. Cost of denormalization
 - b. INSTEAD OF triggers
 - i. Nuances of how triggers are being fired
 - ii. Dangers of composite primary keys
 - iii. Case Study: Handling Updates - Dynamic SQL vs. Full-Table Update
6. Caching techniques:
 - a. Built-in caching
 - i. DETERMINISTIC clause
 - ii. Sub-query caching
 - iii. RESULT CACHE
 - b. Manual caching mechanisms
7. Tips, tricks, discussions.

Objectives:

1. Introduce attendees to PL/SQL performance tuning techniques
2. Extensively cover the most efficient ways of communicating between SQL and PL/SQL
3. Illustrate how PL/SQL can be used to solve complex production problems