Legacy PL/SQL – what is it?

A REVIEW OF THE PL/SQL SITUATION SINCE THE FIRST RELEASE IN 1988. BILLIONS OF PL/SQL PROGRAMS HAVE BEEN DEVELOPED SINCE THEN AND MOST ARE STILL IN USE.

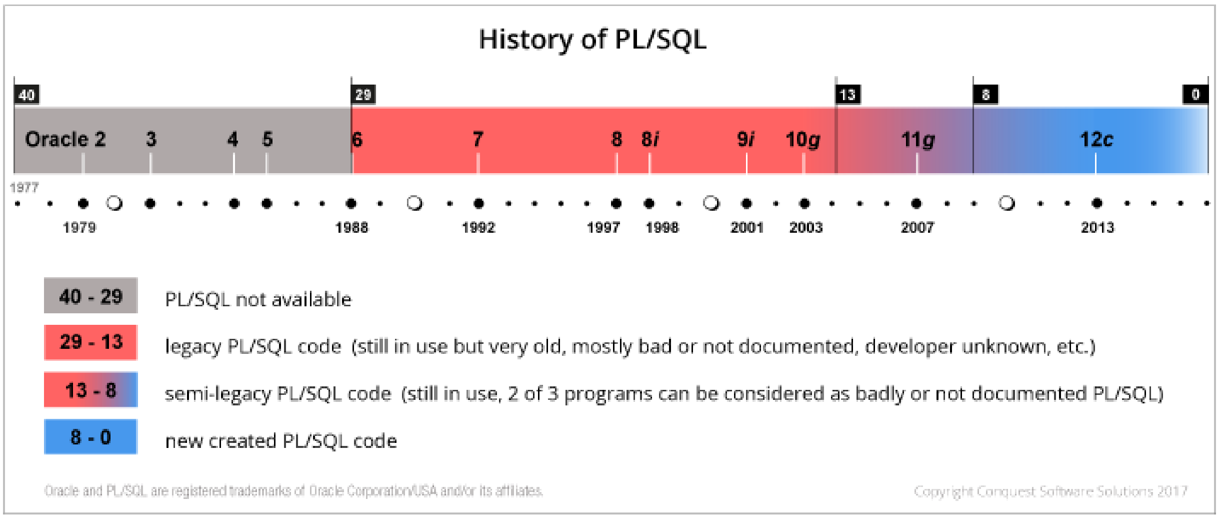
1. Introduction to PL/SQL (**P**rocedural **L**anguage/**S**tructured **Q**uery **L**anguage)

PL/SQL is Oracle’s procedural language extension to SQL. The key strength of PL/SQL is its tight integration with the Oracle database. PL/SQL is the most common language for the world of Oracle databases. Developers are realizing the benefits in both application performance and database performance by implementing the database interaction in PL/SQL. In addition PL/SQL is portable and platform independent within the Oracle database family and runs on all supported Oracle platforms.

2. Facts about PL/SQL

- 1988 - PL/SQL released
- 350.000 Oracle database customers (~ user of PL/SQL)
- 3.6 million PL/SQL developers (approx.)
- 1.2 million Oracle DBAs (approx.)

3. Legacy PL/SQL - what is it?



Generally speaking legacy PL/SQL refers to code inherited by a team or an individual programmer from somebody else (externally or internally). PL/SQL code older than eight to nine years is considered legacy PL/SQL code.

Frequently the original developer is not available or known anymore and/or the code is badly or not documented at all. In many cases the processes run in the stored programs are not understood in detail. These situations are difficult to support and maintain.

IT departments are reluctant to touch legacy code for fear of opening Pandora’s box. The lack of knowledge prevents technical progress and cost efficiency.

Running mission critical applications it represents a constant risk. "*Don't touch anything which is not broken*" is a common strategy. A very unsatisfactory and risky situation CIOs of large enterprises face globally.

4. Why PL/SQL and not Java? The war of opinions…

In the late 1990s Oracle started to promote Java as the main programming language for its database technology. Compatibility, portability, ease of learning and high performance were the main arguments. Oracle seemed to abandon the PL/SQL world.

25 years later PL/SQL is still in use – not only to maintain existing applications but also to generate new database scripts and programs - and the demand for PL/SQL has not decreased. Databases are long-term company assets and represent a significant value of money spent and accumulated knowledge. Extending the useful life of existing PL/SQL stored programs helps to protect this investment.

PL/SQL stored programs are at the core of Oracle databases. They directly influence the performance, functionality, stability and security of Oracle database applications. PL/SQL has become – and will remain for a long time - a mission critical element in all professional IT environments based on Oracle database technology.

5. PL/SQL - the choice of Oracle DB developers today and in the future

- Main arguments in favor of PL/SQL versus Java are better database performance and less network traffic
- PL/SQL is best used to access the database. Java can be used to handle complex code not accessing the database
- The majority of existing "stored programs" and "in-database code" for Oracle databases is written in PL/SQL
- Java is object oriented. The Oracle database was developed solely on the relational model. This causes various problems using Java in the Oracle DB environment. From v9 Oracle supports the object oriented model, however, the core database is still based on the relational model. Object and relational methodologies do not mix well

6. *Clear*SQL and *Clear*DB Documenter

Conquest database tools help Oracle users to support and maintain legacy PL/SQL code, keep it operational, and warrant 24/7 availability.

Conquest tools address the "Oracle Developer and Database Administration" market. They are a powerful set of tools for the so called "legacy PL/SQL" market and help develop, maintain, and keep Oracle legacy PL/SQL code alive.

Currently billions of PL/SQL stored programs are running in Oracle databases all over the world. They are key to keep the databases up and running. Quantifying their value in money is impossible. Developing them was a huge investment in terms of money and knowhow. Replacing them will not only be expensive but can also be risky for the reasons mentioned before.

In times of continuing cost pressure and required cost cuts CIOs face the challenge of reducing the maintenance costs of their PL/SQL environments while ensuring 100% availability of their Oracle databases.

Well-known companies such as **T-Mobile, General Dynamics, Deutsche Bank, Moody's, Walt Disney, Telecom Italia, Emerson, Sony, Lockheed-Martin, Bank of China, European Commission, State of New Jersey, Sony, EATON, Merrill Lynch, Northrop Grumman, The Boston Globe, Allianz, US Department of Justice, Yahoo!, Booz Allen Hamilton, Banchile, Daimler AG, Cap Gemini, GE, Accenture,** to name just a few, purchased *Clear*SQL and/or *Clear*DB Documenter and/or SQL*Detective* to maintain their PL/SQL code and their Oracle databases and keep them up and running (24/7). At the same time they reduce maintenance costs and remove the anxiety over undocumented code. A very attractive combination.

*Clear*SQL and *Clear*DB Documenter offer new features and functions currently not provided by any other competitive product. They are the only Oracle DB tools capable of creating "clickable" diagrams of PL/SQL code; e.g. Flowcharts (logic), Call Tree diagrams (data flow) and CRUD matrices (relationship between processes and the data they use).

*Clear*SQL is the leading PL/SQL code analyzer for analyzing, illustrating, maintaining, and supporting Oracle legacy PL/SQL code. A powerful code review and quality control tool for Oracle PL/SQL. *Clear*SQL creates a series of industry accepted quality control analysis of PL/SQL code to identify potential coding problems. Its code review features provide recommendations for better coding style and check for error-prone areas.

*Clear*DB Documenter, the unique Oracle database documenter provides an easy way to document all Oracle databases a DBA manages. It documents 59 Oracle database object types (schema and non-schema) up to Oracle 12c. It updates previously generated documentation to ensure that the documentation is always up-to-date and synchronized with the database. It compares a documentation with its Oracle DB and reports the changes in the DB since the documentation was created. *Clear*DB Documenter, using its integrated engine of *Clear*SQL, analyzes and documents PL/SQL objects, their source codes, status, references and dependencies. To complete its documentation of PL/SQL, it creates "clickable" Flowcharts, Call Tree diagrams and CRUD matrices from PL/SQL code.

The Database Security Audit is the latest product innovation of *Clear*DB Documenter and addresses the increasingly important concern about data protection. The extensive, „built-in“ Oracle Database security audit checks more than 600 potential security risk aspects of the Oracle Database (up to 12c). A detailed report summarizes all failed checks by severity and provides guidance to resolve the security issues identified.