

ORACLE®

Why use PL/SQL?

Bryn Llewellyn
Distinguished Product Manager
Database Division
Oracle HQ
twitter: @BrynLite

January 2016

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

BIWA Summit 2016 – YesSQL

Large software systems must be built from modules. A module hides its implementation behind an interface that exposes its functionality. This is computer science's most famous principle. For applications that use an Oracle Database, the database is the persistence module. The tables and the SQL statements that manipulate them are the implementation details. The interface is expressed with PL/SQL. Developers and users of applications built this way are happy with their correctness, maintainability, security, and performance. But when developers follow the NoPlsql paradigm, their applications have problems in each of these areas and users suffer. This session provides PL/SQL devotees with unassailable arguments to defend their beliefs against attacks from skeptics. Skeptics who attend will see the light.

Whitepaper download

- My presentation slides are no use without me presenting them
- I have written a whitepaper that gives a careful and detailed account of what my presentation addresses
 - go to the PL/SQL page on OTN: **`oracle.com/plsql`**
 - notice the link to the Oracle Database PL/SQL and EBR blog
 - notice the blog that announces the “Why use PL/SQL?” whitepaper
 - notice my Twitter handle *@BrynLite*.
I’ll tweet news on updates to the paper

Hardware and Software Engineered to Work Together